Saskatchewan
Research Council

TAILS ANALYSIS (A METHOD FOR THE STUDY OF ELEMENT ASSOCIATIONS IN "ANOMALOUS" CONCENTRATION RANGES): METHOD, APPLICATIONS, AND A COMPUTER PROGRAM

by

Michel Mellinger

SRC Technical Report No. 148

Publication R-740-4-B-83

December, 1983

TAILS ANALYSIS (A METHOD FOR THE STUDY OF ELEMENT ASSOCIATIONS IN "ANOMALOUS" CONCENTRATION RANGES): METHOD, APPLICATIONS, AND A COMPUTER PROGRAM

by

Michel Mellinger

SRC Technical Report No. 148

R-740-4-B-83

December, 1983

TAILS ANALYSIS (A METHOD FOR THE STUDY OF ELEMENT ASSOCIATIONS IN "ANOMALOUS" CONCENTRATION RANGES): METHOD, APPLICATIONS, AND A COMPUTER PROGRAM

by

Michel Mellinger

SRC Technical Report No. 148

December, 1983

SRC Publication No. R-740-4-B-83

TABLE OF CONTENTS

		Page
INTRODUCTION		1
I.	TAILS ANALYSIS : THE METHOD	3
.1	Definition of the Problem	3
.2	Element Associations in Distribution Tails	5
.3	Robustness, Flexibility, and Operating Rules	11
.4	Formalization	16
.5	Extensions	18
II.	EXAMPLES OF APPLICATION OF TAILS ANALYSIS	20
.1	Element Associations in Lake Sediments	20
.2	Lithogeochemical Signature of a Hydrothermal Alteration Process	24
.3	Element Associations and Mineralization Process(es)	29
III.	THE TAILAN COMPUTER PROGRAM	38
.1	Purpose	38
.2	General Structure and Input/Output	38
.3	Memory Management	42
.4	Input Parameters	45
.5	Printout	47
.6	Output Files	50
.7	Increasing the Memory Vector	51
.8	Modifying the Data Conversion Routine	52
APPENDIX A - Listing of the Computer Program TAILAN		55
APPENDIX B - Example of Printout		85

List of Tables

		Page
Table 1.	Analytical data.	6
Table 2.	Normalized tail associations for six trace elements analyzed in lake sediments from three adjacent drainage basins.	22
Table 3.	Normalized tail associations for 13 trace elements analyzed in clay-altered rocks derived from different rock types, from the Maurice Bay area (Saskatchewan).	26
Table 4.	Element tail-scores for the three groups of samples identified in Figure 8.	33
Table 5.	Sample distribution according to element tail-scores and initial rock type.	33
Table 6.	Number of tails observed in two sample groups, according to elements and initial rock type.	33
Table 7.	Normalized tail associations for 11 elements analyzed in clay-altered rocks from the Midwest Lake property.	35
Table 8.	Tasks of the various routines of TAILAN.	40
Table 9.	Input/Output file requirements of TAILAN	41
Table 10.	Content and usage of COMMONs in TAILAN	41

List of Figures

		Page
Figure 1.	Examples of frequency distributions for element concentrations	4
Figure 2.	Frequency distributions.	6
Figure 3.	Tails analysis applied to the data of Table 1 (first example).	8
Figure 4.	Tails analysis applied to the data of Table 1 (second example)	13
Figure 5.	Boundary cases for the Tails Table	15
Figure 6.	Element associations for six trace elements analyzed in lake sediments from three adjacent drainage basins	23
Figure 7.	Element associations for 13 trace elements analyzed in clay-altered rocks derived from different rock-types, from the Maurice Bay area (Saskatchewan)	27
Figure 8.	Histogram of the sample tail-scores for clay-altered samples from the Midwest Lake property (Saskatchewan).	31
Figure 9.	Element associations for 11 elements analyzed in clay-altered samples from the Midwest Lake property	36
Figure 10.	Interpretation of the element associations from Figure 9.b in terms of ore minerals and mineralization events	36
Figure 11.	General Structure of TAILAN	39
Figure 12.	Input parameter cards	46

INTRODUCTION

Tails Analysis was developed in 1981 as a means of studying element associations in mineralized and anomalous rock samples from the Maurice Bay area in the Athabasca Basin (Mellinger, 1981). It was later applied to various geochemical data, and formalized.

Tails Analysis is a method for the study of element associations in "anomalous" concentration ranges, i.e., in the tails of concentration frequency distributions. It allows to go further in the study of element associations than classical methods which use linear correlation coefficients, because it uses robust similarity coefficients based on a binary (0,1) re-coding of the data. Flexibility is also built into the method, as the user is responsible for defining the parameters of the re-coding procedure. Finally, complex situations can be examined flexibly at the interpretation stage.

This report consists of three chapters:

Chapter I is a presentation of Tails Analysis. It first uses a simple example to explain the various steps of the method and their meaning. Various aspects of the method are then discussed, and a formal presentation of Tails Analysis follows.

Chapter II illustrates the usefulness of Tails Analysis by using three examples of application. In the first example, element associations in lake sediments for adjacent drainage basins are studied. In the second example, the lithogeochemical signature left by a hydrothermal process on two initial rock types is examined. In the third example, a complex situation is examined, which involves superimposed mineralization events affecting various rock types.

Chapter III presents the TAILAN computer program. Information is provided about its programming and usage. A listing of TAILAN is provided in Appendix A, and an example of printout in Appendix B. A tape copy of TAILAN is available on request.

NOTE: in a first reading, the reader may skip sections I.4 and I.5 from Chapter I. In addition, sections III.2 and III.3 from Chapter III are concerned with the programming of TAILAN more than with Tails Analysis itself, and may thus be ignored by those readers who have limited interest in this subject.

I. TAILS ANALYSIS : THE METHOD

I.1. Definition of the Problem

Let us assume that a series of samples were collected and chemically analyzed for several elements. The samples may be regional lake sediments, rock samples from a particular property, or any other type of geochemical samples.

The distribution of concentrations for the elements may correspond to various types of populations, but will commonly contain some values that are notably greater than others, that will appear on histograms as separate sub-populations or as distribution tails (Figure 1). Samples containing high concentrations of one or more elements are the (univariate) anomalous samples sought by the exploration geochemist.

In addition to simply considering each element individually, one may ask the following question: are the elements associated in any systematic way for a set of anomalous samples? This question implies:

(1) that one is interested in the set of anomalous samples in particular (an anomalous sample is a sample which contains an anomalous concentration of one element at least);

(2) that one is interested in element associations: the answer will be at the multivariate level, and "anomalies of association" may thus be detected that were not obvious at the univariate level.

Tails Analysis is a simple non-parametric method based on a binary re-coding of the chemical data, that easily produces answers to the previous question.

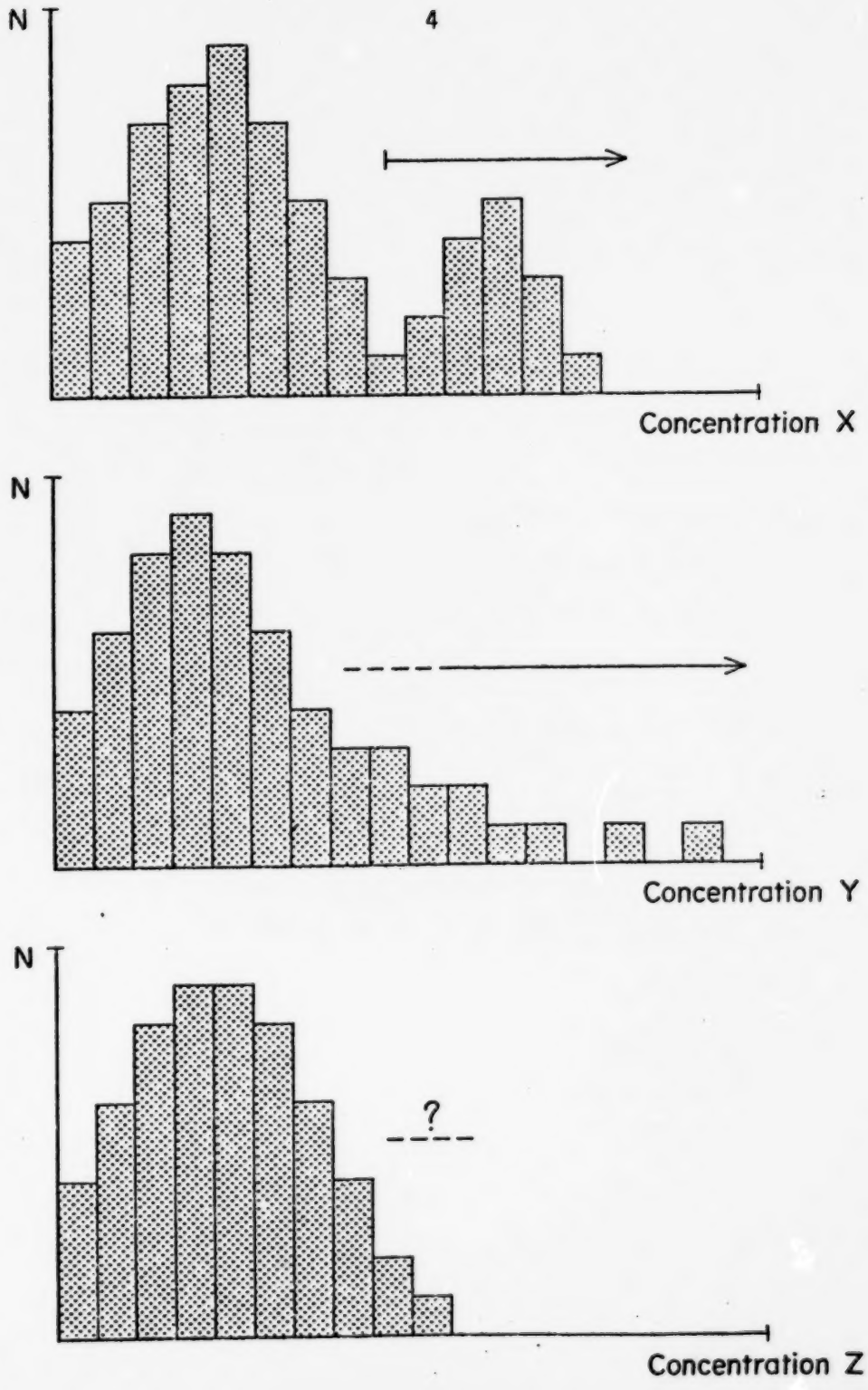


Figure 1. Examples of frequency distributions for element concentrations: (X) distinct sub-populations, (Y) distribution tail, (Z) no distinct distribution tail.

I.2 Element Associations in Distribution Tails

Tails Analysis was developed as a means of studying element associations in distribution tails. The procedure that is used will be described here using a simple data table consisting of 10 samples analyzed for 3 elements: Cu, Zn, and U (Table 1). A formal presentation of the method is given in Section I.4 below.

Frequency distributions for the 3 elements are shown in Figure 2. In Figure 2.b, 3 samples are identified in order to illustrate the question that was stated in the previous section and that we want to answer. This question is also: are higher concentrations of these 3 elements Cu, Zn, and U, found in the same samples, or are they present independently of each other? The procedure for Tails Analysis is as follows:

First step: define "anomalous concentrations".

Anomalous concentrations are defined as those concentrations that are higher than a given value selected by the geochemist. This is the classical univariate approach. The selected value, or threshold or cut-off value, may be chosen according to various rules that will depend on the shape of the frequency distribution curve and the geochemist's experience. The cutoff value for an element may be the value that discriminates between two sub-populations; it may also discriminate between a "main" population and a distribution tail in the case of a positively skewed total population; it may also be quite arbitrarily chosen, in particular when the frequency distribution is continuous and only slightly skewed. Other considerations will be discussed in the next section.

	Cu	Zn	U
Sample # 1	12	56	1
Sample # 2	5	42	4
Sample # 3	10	45	2
Sample # 4	13	52	5
Sample # 5	8	38	7
Sample # 6	25	115	2
Sample # 7	18	85	2
Sample # 8	5	40	9
Sample # 9	8	53	1
Sample #10	15	65	1

Table 1 - Analytical data (all units are ppm)

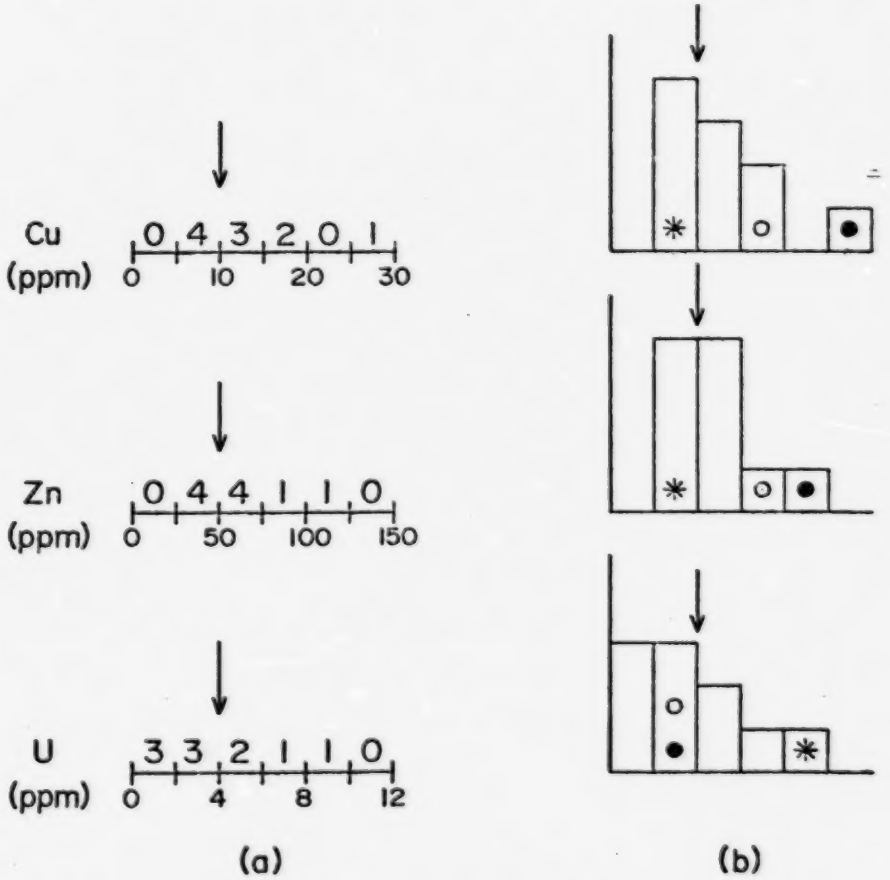


Figure 2. Frequency distributions. Same horizontal scales on (a) and (b); arrows indicate cutoff values; see text for discussion.

In our example (Figure 2), the cutoff values for Cu, Zn, and U are respectively: 10 ppm, 50 ppm, and 4 ppm.

Second step: re-code the initial data using binary coding.

The initial data are re-coded in the following way, for each element:

IF initial concentration < cutoff value, THEN code value = 0;

IF initial concentration ≥ cutoff value, THEN code value = 1.

The result is a data table containing only 0's and 1's, the tails table. A "1" in the table indicates that the sample corresponding to this row of the table has a tail concentration for the element in the corresponding column. The tails table is thus a presence/absence data table. Further characteristics of the table are summarized as marginal frequencies: in the right margin appears the number of tails observed for each sample or sample tail-scores, and in the bottom margin appears the number of tails observed for each element, or element tail-scores.

For our example, the tails table is shown in Figure 3.a. Cu has 6 tails, so does Zn, and U has 4 tails (N_t); 5 samples have 1 tail, 4 samples have 2 tails, and 1 sample has 3 tails (N_s). The total number of tails is 16.

Third step: calculate element associations in tails.

The tails table is a presence/absence table, and element associations in tails can simply be derived from it by counting:

(1) how many times elements A and B happen to have tail concentrations together; this is done for all possible pairs of elements;

(2) how many times each element has a tail concentration by itself.

initial data as in Table 1

↓

cutoff values (ppm):

Cu(10), Zn(50), U(4) →

Cu	Zn	U	N_s:
1	1	0	2
0	0	1	1
1	0	0	1
1	1	1	3
0	0	1	1
1	1	0	2
1	1	0	2
0	0	1	1
0	1	0	1
1	1	0	2
N_t: 6	6	4	(16)

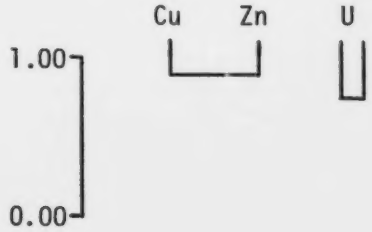
3.a - tails table

	Cu	Zn	U	N_t	N_{ta}
Cu	1	5	1	6	7
Zn		1	1	6	7
U			3	4	5

3.b - tail associations table.

	Cu	Zn	U	N_t	N_{av}
Cu	.17	.83	.25	6	1.17
Zn		.17	.25	6	1.17
U			.75	4	1.25

3.c - normalized tail associations table.



or Cu═Zn U

3.d - graphical representation of the element associations

Figure 3. Tails Analysis applied to the data of Table 1 (first example).

These counts are tail associations, and are compiled in the tail associations table which is a symmetrical table elements x elements. Characteristics of interest that are added to this table are the tail-scores for the elements (bottom margin of the tails table - see second step), and the number of tail associations for each element, which is the margin of the tail associations table just calculated. The number of tail associations for each element will be greater than or equal to the tail-score for this element.

For our example, the tail associations table is shown in Figure 3.b. Only one half needs to be displayed as the table is symmetrical. Along the diagonal of the table are found the "isolated tails" for each element: Cu, Zn, and U have a tail by themselves respectively: once, once, and three times. Off-diagonal counts show that Cu and Zn have tails together five times, whereas U shares tails once with Cu and with Zn. Finally, the number of tail associations for each element (N_{ta}) is respectively equal to 7, 7, and 5.

Now that tail associations have been calculated, one can try to answer the initial question. A quick study of the tail associations table in this case shows that Cu and Zn are systematically associated, whereas U is poorly associated with either of these two elements and is thus essentially anomalous by itself.

One could consider the tail associations table as the answer that was sought. However, when several elements are involved for a large number of samples, the direct interpretation of tail associations may be biased for two reasons:

(1) the values of the tail associations are directly related to the number of samples, and, as a consequence, varying the cutoff values will produce successive tail associations of varying intensity, making comparisons difficult for the same set of samples;

(2) the values of the tail associations are bound by zero (as a minimum) and by the tail-score for each element (as a maximum for tail association A-B: the smallest of the tail-scores for element A and element B).

It is thus necessary to normalize the tail associations.

Fourth step: normalize tail associations.

The maximum value that can be obtained for the tail association A-B between elements A and B is equal to the tail-score for element A or element B, whichever is smaller. Normalization of the tail associations can thus be carried out simply by dividing each tail association by this maximum expected value; the set of values necessary for this normalization was calculated previously and is the bottom margin of the tails table.

The result is a set of normalized tail associations, grouped in the normalized tail associations table. The tail-score for each element is kept as a reminder of the normalization coefficients that were used to calculate the new table; the average number of tail associations per sample involved is also calculated for each element. Normalized tail associations have values between 0.00 and 1.00; the latter value would indicate maximum possible association between two elements, or maximum "isolation" of one element if occurring on the diagonal of the table.

Normalized tail associations are thus similarity coefficients between the elements, or correlation coefficients in terms of co-occurrence in distribution tails. They can be interpreted by clustering first those elements that have higher similarity, following classical cluster analysis aggregation rules or more empirical procedures. The number of tails for each element shall also be considered when grouping elements, in particular when wide variations are present between elements (see Section I.3 for further discussion).

For our example, the normalized tail associations table is shown in Figure 3.c. The tail-score for each element (N_t) is repeated as these were used to calculate the similarity coefficients of the table. The average number of tail associations per sample involved (N_{av}) varies from 1.17 to 1.25. The interpretation of the element associations are expressed graphically in Figure 3.d, using both a classical dendrogram and a more general element association diagram.

I.3 Robustness, Flexibility, and Operating Rules

The binary coding used in Tails Analysis is a robust coding scheme, that is it is not sensitive to variations in the data. For example, for a cutoff value of 10 ppm for Cu, any concentration lower than 10 ppm is re-coded as "0"; similarly, any concentration greater than or equal to 10 ppm is re-coded as "1". Another advantage is that essentially the same tails table is obtained from a set of initial data when cutoff values are slightly increased or decreased. Or, the selection of cutoff values used need not be very precise with respect to the range of concentration values, as similar tails will be obtained within a certain range of cutoff values for each variable.

Tails Analysis calculations follow a set procedure after the cutoff values have been selected. However, flexibility is built into the method at its first step, when the selection of cutoff values is made. By repeating the procedure on different sets of cutoff values, the user can "explore" the nature of tail associations and the manner in which they vary for various definitions of the anomaly thresholds. This permits in particular to examine and test the robustness of the tail associations observed. A second example is given in Figure 4, where the same initial data that were used in the previous section are transformed into a new tails table on the basis of the following cutoff values: Cu (15 ppm), Zn (75 ppm), and U (6 ppm); all cutoff values are thus increased with respect to the first example, presented in Figure 3. The number of tail associations now is much less than previously, as could be expected; their nature, however, is very similar as witnessed by the element association diagrams.

Such flexibility will be exercised between two boundary cases:

(1) the case when cutoff values are so low that the tails table contains too many, or even only, 1's: very little or no useful information is then present in the tails table (Figure 5.a); the tails table can be considered as saturated, as the tail-scores (marginal frequencies) become close or equal to the number of variables (right margin) or to the number of samples (bottom margin).

(2) the case when cutoff values are so high that the tails table contains very few 1's and in particular very few overlaps between tails, i.e., very few tail associations (Figure 5.b);

initial data as in Table 1

↓

cutoff values (ppm):

Cu(15), Zn(75), U(6) →

Cu	Zn	U	N_s:
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	1	1
1	1	0	2
1	1	0	2
0	0	1	1
0	0	0	0
1	0	0	1
N_t: 3	2	2	(7)

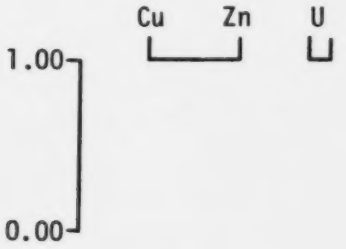
4.a - tails table

	Cu	Zn	U	N_t	N_{ta}
Cu	1	2	0	3	3
Zn		0	0	2	2
U			2	2	2

4.b - tail associations table

	Cu	Zn	U	N_t	N_{av}
Cu	.33	1.00	-	3	1.00
Zn		-	-	2	1.00
U			1.00	2	1.00

4.c - normalized tail associations table.



or Cu═Zn U

4.d - graphical representation of the element associations.

Figure 4. Tails Analysis applied to the data of Table 1 (second example). Compare with the first example (Figure 3).

initial data as in Table 1

↓

cutoff values (ppm):

Cu (5), Zn (40), U (2) →

Cu	Zn	U	N_s:
1	1	0	2
1	1	1	3
1	1	1	3
1	1	1	3
1	0	1	2
1	1	1	3
1	1	1	3
1	1	1	3
1	1	0	2
1	1	0	2
N_t: 10	9	7	(26)

5.a - saturated tails table: cutoff values very low

initial data as in Table 1

↓

Cutoff values (ppm):

Cu (35), Zn (100), U (7) →

Cu	Zn	U	N_s:
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	1	1
0	1	0	1
0	0	0	0
0	0	1	1
0	0	0	0
0	0	0	0
N_t: 0	1	2	(3)

5.b - almost-empty tails table: cutoff values very high

Figure 5. Boundary cases for the Tails Table.

the tails table is then almost empty, and the tail-scores for the samples are mostly equal to 0 or $\bar{1}$ (right margin).

Useful cases will thus be characterized by a "useful level of tail associations". No general criterion has been developed to measure such usefulness or optimal level of tail associations, so the user must rely on his own good judgement. The user should examine the marginal frequencies of the tails table as a guide to the selection of cutoff values and to the interpretation of tail associations:

(1) examination of the distribution of the sample tail-scores (right margin of the tails table) gives an indication of the richness and consistency of the distribution of tails among the samples (see Section II.3);

(2) examination of the tail-scores for the elements (bottom margin of the tails table) shows how consistently the cutoff values define distribution tails in terms of the number of samples involved in each of them. In particular, during the interpretation of normalized tail associations, it is important to take these numbers into consideration when aggregating the elements: the "main structure" of element associations should involve those elements that have larger distribution tails, whereas those elements that have small distribution tails should only be appended to the "main structure" obtained (see examples in sections II.2 and II.3). Indeed, clustering elements on the basis of the normalized tail associations only may lead to unstable element associations when some of the elements have distribution tails significantly smaller than the other elements.

I.4 Formalization

NOTE: reading this section is not required for understanding and using Tails Analysis. This section is written for those readers who:

(1) are interested in a more formal presentation of the method than was given in Section I.2; or

(2) wish to confirm the definition of some of the entities defined previously; or

(3) wish to follow or modify the computer program presented in the third chapter.

NOTATIONS:

M_{nxp} designates a matrix M with n rows and p columns.

m_{ij} designates the generic element of matrix M, with indices i and j varying according the size of matrix M as defined previously.

$m_{i.} = \sum_{j} m_{ij}$

$m_{.j} = \sum_{i} m_{ij}$

M' designates the transpose of matrix M.

The <u>data table</u> contains n samples analyzed for v elements, and is:

$$D_{nxv} = \{d_{ij}\}$$

The set of <u>cutoff values</u> for the v elements is:

$$C_{1xv} = \{c_j\}$$

The <u>tails table</u> is defined as:

$$T_{nxv} = \{t_{ij}\}$$

with: $t_{ij} = 0$ if $d_{ij} < c_j$

$t_{ij} = 1$ if $d_{ij} \geq c_j$

The right margin of the tails table contains the sample tail-scores, and is:

$$TK_{nx1} = \{t_{i.}\}$$

The bottom margin of the tails table contains the element tail-scores, and is:

$$TL_{1xv} = \{t_{.j}\}$$

The tail associations table A can be calculated as A=T'.T. However, the diagonal elements of T'.T are the elements of TL defined above. As we wish to define the diagonal elements of A as the set of number of isolated tails encountered in the tails for the elements, A is defined as:

$$A_{vxv} = \{a_{jj'}\}$$

with: $a_{jj'} = \sum_k t'_{jk} \cdot t_{kj'}$ $\quad ,j' \neq j$

$$a_{jj} = n_j = \sum_i \delta_i \cdot t_{ij}$$

with $\begin{cases} \delta_i = 1 & \text{if } t_{i.} = 1 \\ \delta_i = 0 & \text{otherwise} \end{cases}$

Note: A is symmetrical: $a_{j'j} = a_{jj'}$

The margin of the tail associations table contains the number of tail associations for each element, and is:

$$AK_{1xv} = \{a_{j.}\} = \{a_{.j'}\}$$

The normalized tail associations table is defined as:

$$A^*_{vxv} = \{a^*_{jj'}\}$$

with: $a^*_{jj'} = a_{jj'}/\text{Min}\,(t_{.j}, t_{.j'})$

Note: A* is symmetrical: $a^*_{j'j} = a^*_{jj'}$

Note: because $a_{jj'} \leq \text{Min}\ (t_{.j}, t_{.j'})$

and $a_{jj} = n_j \leq t_{.j}$,one has

$0.0 \leq a^*_{jj'} \leq 1.0 \quad \forall j,j'$. This means that the $a^*_{jj'}$ can be considered as similarity coefficients.

Finally, the set of <u>average number of tail associations per sample involved</u> is defined as:

$$S^*_{1xv} = \left\{s^*_j\right\}$$

with: $s^*_j = a_{j.}/t_{.j}$

Note: S* can be calculated from A* and TL, by first calculating A: $a_{jj'} = a^*_{jj'}.\ \text{Min}\ (t_{.j}, t_{.j'})$, for all j'; then by using the definition of s^*_j above. This procedure will be used when S* must be calculated using only information saved on file #35 by computer program TAILAN (see Section III.6).

I.5 Extensions

Tails Analysis is based on a binary re-coding of the initial data, which produces the tails table. The similarities between the elements are then studied using a particular similarity coefficient, the normalized tail association defined formally as $a^*_{jj'}$ in the previous section.

If this similarity coefficient suits our need here, it is possible that the geochemist may be interested in studying element associations in distribution tails using some other similarity coefficient calculated from the presence/absence codes contained in the tails table, and suiting a different need.

Also, it may be of interest to investigate element associations or sample associations using other classification methods that can treat presence/absence data, in which case the tails table may be used as input to such methods. The computer program TAILAN, presented in Chapter III, allows saving of the tails table on a computer storage medium.

II. EXAMPLES OF APPLICATION OF TAILS ANALYSIS

In this chapter, three examples of application of Tails Analysis are presented. In the first example, differences in element associations between sub-areas of a regional lake sediment geochemical survey are investigated. In the second example, the geochemical signature of a hydrothermal process is characterized for two different initial rock types that were affected; this example also shows how element associations involving various "weights" may be approached. The third example illustrates how various element associations may be superimposed in a tails table and how such a particular situation may be interpreted.

II.1 Element Associations in Lake Sediments

The data come from a regional geochemical survey that covered part of the eastern portion of the Athabasca Basin (Saskatchewan). The lake sediment samples were analyzed for six elements: U, Zn, Cu, Pb, Ni, and As. A total of 440 samples are considered here. The total population was divided into three groups, each related to a different drainage basin. The three basins and their corresponding populations are: basin A (331 samples), basin B (46 samples), and basin C (63 samples).

The cutoff values were selected after examination of the total population; they are:

U: 3 ppm, Zn: 60 ppm, Cu: 12 ppm,

Pb: 2 ppm, Ni: 10 ppm, As: 2 ppm.

The size of the tail population for each element varies from 21% to 36% of the total population.

Normalized tail associations were calculated for each group of samples. The results are shown in Tables 2.a to 2.c; element associations in tails are represented graphically in Figure 6.

Interesting similarities and differences in element associations appear from one basin to another. In basins A and B (Figures 6.a and 6.b), common associations are the "skeleton" formed by Zn, Cu, Ni, and also U. However, the "links" between Pb and As and these four elements are very different in basin A compared to basin B: Pb is not noticeably associated with any other element in basin A, whereas it is associated with both Cu and Ni (in addition to being self-associated) in basin B; As is associated with Zn and Ni (slightly only with U) in basin A, whereas it is associated with U (slightly only with Zn) in basin B. As for basin C (Figure 6.c), element associations are much weaker than in the other two basins: Pb is essentially isolated; Cu and Zn still form a strong association; U is associated to some extent with Ni and As; the association between Ni and Cu and Zn can still be detected in the normalized tail associations (Table 2.c), but it is much weaker than previously.

These variations in element associations, within anomalous concentration ranges, from one basin to another cannot be explained by inter-basin variations in the frequency of anomalous samples for each element, as can be seen by comparing the data in Tables 2 (columns N_t and $N_{\%}$ to the right of each table). In terms of geochemical "anomalies", even if some inter-basin variations in the frequency of "anomalous" samples are observed for the elements, relative to the number of "anomalous" samples in each basin (see $N_{\%}$), the nature of the element associations within the "anomalous" population does not vary in the same way.

Basin A	U	Zn	Cu	Pb	Ni	As	N_t	$N_{\%}$
U	.13	.52	.54	.20	.67	.45	80	34
Zn		.20	.71	.36	.61	.62	120	51
Cu			.06	.32	.78	.30	69	29
Pb				.36	.30	.20	81	35
Ni					.14	.52	109	47
As						.13	71	30

2.a - basin A (N=331; 234 anomalous samples).

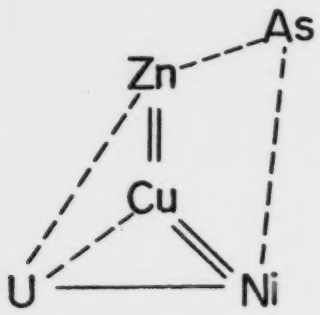
Basin B	U	Zn	Cu	Pb	Ni	As	N_t	$N_{\%}$
U	.13	.38	.25	.38	.63	.50	8	20
Zn		.27	.88	.40	.55	.45	15	38
Cu			.13	.63	.75	.25	8	20
Pb				.54	.64	.18	24	60
Ni					-	.36	11	28
As						.27	11	28

2.b - basin B (N=46 samples; 40 anomalous samples).

Basin C	U	Zn	Cu	Pb	Ni	As	N_t	$N_{\%}$
U	.25	.42	.33	.33	.50	.58	12	23
Zn		.16	.88	.36	.46	.25	25	47
Cu			-	.41	.46	.13	17	32
Pb				.32	.31	.38	25	47
Ni					.23	.38	13	25
As						.31	16	30

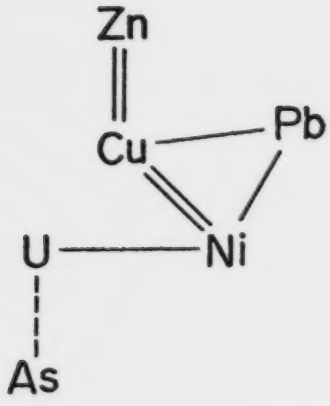
2.c - basin C (N=63 samples; 53 anomalous samples).

Table 2. Normalized tail associations for six trace elements analyzed in lake sediments from three adjacent drainage basins. See text for discussion. See Figure 6 for element associations. (N% is the proportion of anomalous samples for each of the elements concerned).

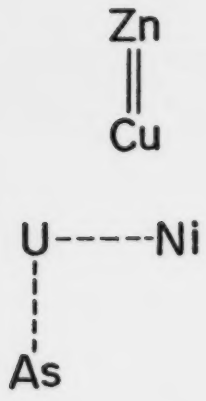


Pb

6.a - basin A



6.b - basin B



Pb

6.c - basin C

Association levels: ═ (≥.70) ── (.60's) - - - (.50's)

Figure 6. Element associations for six trace elements analyzed in lake sediments from three adjacent drainage basins. See text for discussion. See Table 2 for normalized tail associations.

Thus, causes for inter-basin variations in element associations are likely to reflect variations in the physico-chemical characteristics of the lake sediment environment and/or in the nature of these sediments. These results are important both from the viewpoint of the environmental characterization of the lake sediments from the three drainage basins, and from the viewpoint of the exploration geochemist who will need to interpret environmental factors correctly before being able to access "true" anomalies related to the dispersion of elements of interest from a mineralized source.

In this example, Tails Analysis proves to be a simple and effective method for investigating element associations in anomalous concentration ranges.

II.2 Lithogeochemical Signature of a Hydrothermal Alteration Process

The data consist of trace element analyses of drill core samples from the Maurice Bay area (Mellinger, 1981, 1982). All samples have been altered to clay minerals by hydrothermal alteration; 16 samples were originally pelitic gneiss from the basement complex, while 27 samples were sandstone from the Athabasca Group. All samples were analyzed for 15 trace elements: Mn, Cu, Zn, Pb, Ni, Li, Ba, Sr, V, Co, As, Ag, U, Mo, and B.

The cutoff values were selected after examination of total populations from both the basement complex and the Athabasca Group (respectively 160 and 224 samples), and are:

Mn (300 ppm);	Cu (50 ppm);	Zn (100 ppm);
Pb (50 ppm);	Ni (200 ppm);	Li (300 ppm);
Ba (300 ppm);	Sr (50 ppm);	V (300 ppm);

Co (100 ppm); As (10 ppm); Ag (2 ppm);
U (20 ppm); Mo (10 ppm); B (950 ppm).

The cutoff value for B actually corresponds to the concentration level at which measurements were not possible anymore, using emission arc spectrometry, due to saturation of the emission line. It may also be noted that B concentrations obtained with this analytical method are only semi-quantitative.

Normalized tail associations were calculated for each group of samples. The results are shown in Tables 3.a and 3.b. Element associations in tails are represented graphically in Figure 7.

Of the 15 trace elements considered here, Mn and Ba do not show tail concentrations for either of the two types of clay-altered samples, although distribution tails are present for these two elements for the total sample populations. This results from the association of Mn with iron oxides and Ba with phosphates; not with clay minerals. Of the remaining 13 trace elements, only 8 have any tail for clay-altered basement samples (Table 3.a), whereas all are involved in tails for clay-altered Athabasca samples (Table 3.b). Thus, the hydrothermal process responsible for the clayey alteration has left a richer signature in sandstone-derived samples than in gneiss-derived samples.

In gneiss-derived clay-altered samples, element associations can be built around Pb, Sr, V, U, and B, which have more anomalous samples than Cu, As, and Mo (Table 3.a, column N_t). For these 5 elements (Figure 7.a), Sr and V are linked independently (at the 0.75 level) with the B-Pb pair, while U is linked preferentially to V. As for the other elements: Cu is anomalous in one sample along with all previous five

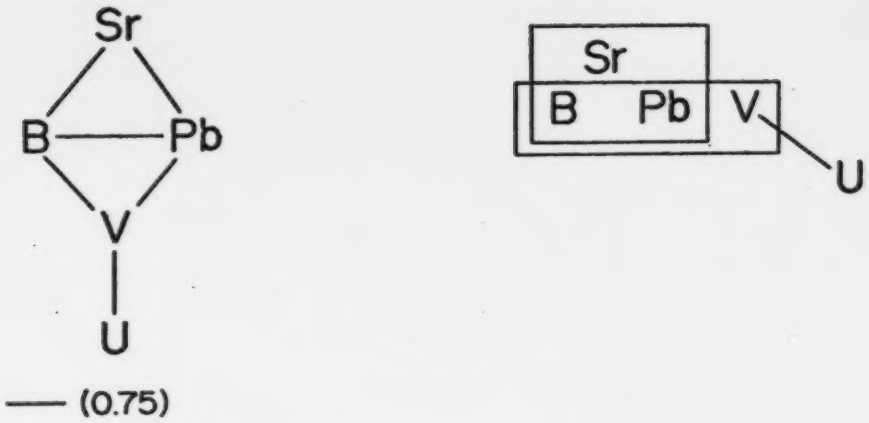
B.C.	Cu	Pb	Sr	V	As	U	Mo	B	N_t
Cu	-	1.	1.	1.	-	1.	-	1.	1
Pb		-	.75	.75	1.	.50	.50	.75	4
Sr			-	.50	1.	.38	.50	.75	4
V				-	.50	.75	-	.75	4
As					-	1.	.50	.50	2
U						-	1.	.50	8
Mo							-	-	2
B								.22	9

3.a - gneiss-derived clay-altered samples (N=16). No sample is anomalous in: Zn, Ni, Li, Co, or Ag.

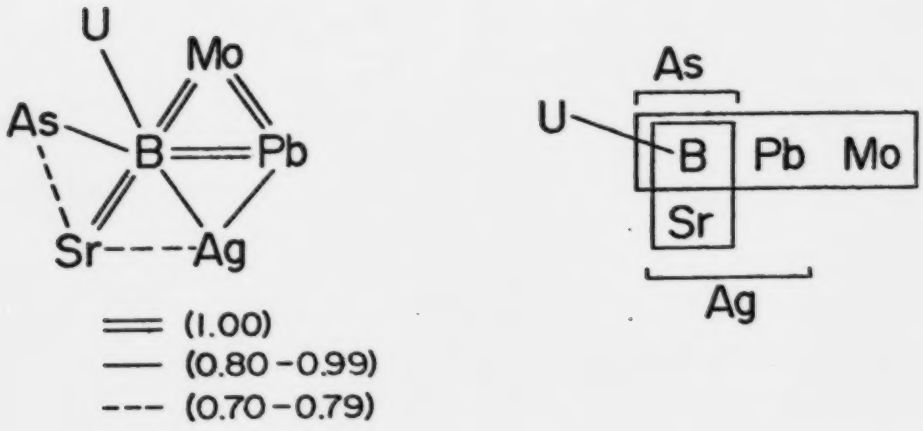
A.G.	Cu	Zn	Pb	Ni	Li	Sr	V	Co	As	Ag	U	Mo	B	N_t
Cu	-	-	1.	1.	1.	1.	1.	1.	1.	-	1.	1.	1.	1
Zn		-	1.	.50	.50	.50	.50	-	.50	.50	1.	1.	1.	2
Pb			-	1.	.75	.66	1.	1.	.27	.86	.55	1.	1.	15
Ni				-	.50	.50	.66	1.	.50	.25	.75	.75	1.	4
Li					-	1.	.66	1.	.75	-	.75	.50	1.	4
Sr						-	.66	1.	.73	.71	.45	.66	1.	15
V							-	1.	.66	-	1.	.33	1.	3
Co								-	1.	-	1.	1.	1.	1
As									-	.43	.45	.66	.91	11
Ag										-	.29	.66	.86	7
U											.09	.50	.82	11
Mo												-	1.	6
B													.08	25

3.b - sandstone-derived clay-altered samples (N=27).

Table 3. Normalized tail associations for 13 trace elements analyzed in clay-altered rocks derived from different rock types, from the Maurice Bay area (Sask.). Mn and Ba, also analyzed, were not anomalous in either group of samples. See text for discussion. See Figure 7 for element associations.



7.a - gneiss-derived clay-altered samples (associations ≥ .75 only).



7.b - sandstone-derived clay-altered samples (associations ≥ .70 only).

Figure 7. Element associations for 13 trace elements analyzed in clay-altered rocks derived from different rock-types, from the Maurice Bay area (Sask.). See text for discussion. See Table 3 for normalized tail associations. Two modes of representation are used; the "bonds" on the left correspond to association levels as defined below each case.

elements; As is relatively more associated to Pb, Sr, and U; finally, Mo is found more systematically with U. However, associations of these three elements with the previous cluster should be interpreted with care as they involve only one or two samples each.

In sandstone-derived clay-altered samples, main element associations involve Pb, Sr, As, Ag, U, Mo, and B, which have more anomalous samples than the six other elements (Table 3.b, column N_t). In this case, a main cluster is formed by the strong association of B, Mo, and Pb, and of Sr with B (Figure 7.b). U, As, and Ag are linked to some elements of this main cluster: U is linked to B, As to both B and Sr, whereas Ag is linked to B, Pb, and also Sr. Of the remaining elements, some are linked to most other elements (Co to all but Ag; Cu to all but Ag and Zn), while others are most specifically associated to the previous cluster: Zn to B, Pb, Mo, and also U; V to B, Pb, and also U; Ni to B and Pb; Li to B and Sr. Here again, Cu, Zn, and Co associations are based on one or two samples and must be considered carefully.

When element associations from the two groups of clay-altered samples are compared, similarities appear despite the "richer" signature observed in the sandstone-derived group. B, Pb, and Sr are all involved in the main cluster in both groups. The V association is generally similar, with V being linked to B and Pb in both groups; this association is not as pronounced in the sandstone-derived group, however. The U association is somewhat different in the sandstone-derived group, where it involves mainly B, compared to the gneiss-derived group, where U is linked mainly to V and less to B. Finally, the most noticeable additional associations in the sandstone-derived group are the

strong association of Mo with B and Pb, and the involvement of significant As and Ag associations with the main cluster.

This example illustrates the usefulness of Tails Analysis for defining geochemical signatures of alteration processes. It also illustrates how normalized tail associations can be interpreted when the elements involved correspond to a wide range of number of anomalous samples. A more robust interpretation is obtained in this case by clustering first those elements which involve a higher number of anomalous samples; in a second step, the other elements can be (tentatively) linked to the observed structure.

II.3 Element Associations and Mineralization Process(es).

The data consist of trace element analyses of drill core samples from the Midwest Lake property (Mellinger and Hoeve, 1982). A total of 40 samples were analyzed, of which two are fresh diabase samples and 38 are clay-altered samples derived from either diabase dykes (11 samples), or sandstone from the Athabasca Group (17 samples), or gneisses from the basement complex (10 samples). All these samples have been extensively altered by hydrothermal alteration and are now essentially comprised of clay minerals. In addition, seven of the clay-altered diabase samples come from an unmineralized area, whereas all other clay-altered samples were collected close to uranium mineralization and are mineralized to various degrees.

We will consider 11 trace elements: Cu, Zn, Pb, Ni, Ba, Sr, V, Co, As, U, and Mo. The purpose of using Tails Analysis here is, in addition to examining element associations, to investigate the possible presence of superimposed mineralization events. The cutoff values were selected on the basis of the total sample population; they are:

Cu (100 ppm);	Zn (50 ppm);	Pb (100 ppm);
Ni (100 ppm);	Ba (100 ppm);	Sr (50 ppm);
V (500 ppm);	Co (100 ppm);	As (10 ppm);
U (50 ppm);	Mo (50 ppm).	

Examination of the tails table shows a relatively large number of tails: 6.6 tails per sample as an average (in the first example, section II.1, the average is of 2.2 tails per sample for 6 elements; in the second example, section II.2, the average is of 2.1 and 3.9 tails per sample for 8 and 13 elements respectively). When a histogram of the sample tail-scores is examined (Figure 8), three groups of samples can be defined: a first group of samples with low tail-scores (equal to 1 or 2), a second with medium tail-scores (between 4 and 7), and a third group characterized by high tail-scores (between 8 and 11).

Which elements are more anomalous in one group than another? The element tail-scores for each group are presented in Table 4. Three observations are of particular interest:

(1) in the first group, in which samples have low tail-scores, only Ni and V are clearly anomalous, with As anomalous in one sample only;

(2) As reaches saturation, i.e., is anomalous in all samples, in both the second and third groups;

(3) Mo approaches saturation in the third group only (it is anomalous in only 2 samples out of 14 in the second group).

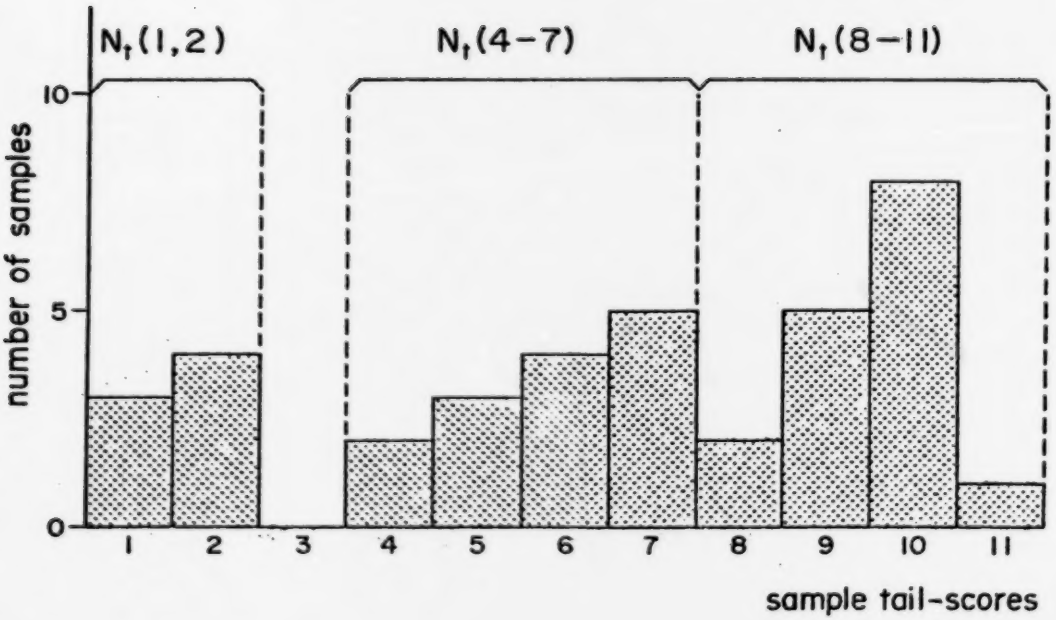


Figure 8. Histogram of the sample tail-scores for clay-altered samples from the Midwest Lake property (Sask.). See text for discussion.

Are samples from one initial rock type more anomalous in one group than another? The number of samples from each initial rock type is presented for each group in Table 5. Most diabase samples have low tail-scores: the seven samples from the first group in fact comprise six diabase samples which, although altered to clay, were collected in an unmineralized area of the property. Samples with medium tail-scores are mostly from the Athabasca Group and the basement complex. Finally, samples with high tail-scores are mostly from the Athabasca Group, indicating that tail saturation was reached more readily in sandstone; this observation is consistent with the fact that sandstone rocks are more porous than basement rocks, and with the related observation that hydrothermal alteration affects a much larger volume of rocks in the Athabasca Group than in the basement complex.

It thus appears that information contained in the tails table is significantly affected by the presence of three initial rock types. The distribution of anomalous samples for elements of interest and according to the three initial rocks types is presented in Tables 6.a and 6.b respectively for the first (low tail-scores) and second (medium tail-scores) groups of samples. For the first group, the dominant role of diabase samples was expected; for the second group, some elements "favor" basements rocks (e.g. Ni, Pb), others "favor" sandstone (e.g. : Sr), while several show anomalous concentration with about equal distribution between sandstone and basement rocks.

Normalized tail associations were calculated for each group of samples. The results are shown in Tables 7.a to 7.c. Because normalized

	Cu	Zn	Pb	Ni	Ba	Sr	V	Co	As	U	Mo	Number of Samples
N_t (1, 2)	-	-	-	4	-	-	6	-	1	-	-	7
N_t (4-7)	5	3	6	11	4	7	9	10	14	11	2	14
N_t (8-11)	13	9	16	16	12	9	16	15	16	16	14	16
Σ	18	12	22	31	16	16	31	25	31	27	16	(37)

Table 4. Element tail-scores for the three groups of samples identified in Figure 8.

	A	B	D	Σ
N_t (1,2)	-	1	6	7
N_t (4-7)	6	6	2	14
N_t (8-11)	11	3	2	16
Σ	17	10	10	(37)

Table 5. Sample distribution according to element tail-scores (vertically-see Figure 8) and initial rock type (horizontally: A=Athabasca, B=basement, D=diabase).

N_t (1,2)	A	B	D	Σ
Ni	-	1	3	4
V	-	-	6	6
As	-	1	-	1

6.a - low tail-scores group

N_t (4-7)	A	B	D	Σ
U	6	4	1	11
Cu	2	2	1	5
As	6	6	2	14
Co	4	5	1	10
Ni	3	6	2	11
Pb	-	4	2	6
Sr	5	2	-	7
V	3	4	2	9

6.b - medium tail-scores group

Table 6. Number of tails observed in two sample groups, according to elements (vertically) and initial rock type (horizontally: A= Athabasca, B=basement, D=diabase). Samples groups after Figure 8.

tail associations are almost completely saturated for the third group (high sample tail-scores), element associations were represented graphically only for the first two groups (Figures 9.a and 9.b).

For the first group (low sample tail-scores) - see Table 7.a and Figure 9.a - Ni and V are the only two elements which are anomalous in more than one sample, and they are associated. It is of interest, however, to notice that V has a relatively high self-similarity (of 0.50): all of the three samples that have a tail-score equal to one are anomalous in V only. The V-Ni association represents a classical geochemical association; the samples in which it is observed are all clay-altered, unmineralized, diabase.

For the second group (medium sample tail-scores) - see Table 7.b and Figure 9.b -, the main element associations involve eight elements. Three elements: Zn, Ba, and Mo, are excluded from the main associations because they involve few samples (low element tail-scores). As, Pb, Ni, Cu, and U are intricately linked in three overlapping clusters: Cu-As-Ni, U-Cu-As, and Pb-Ni-As. Co and Sr are both strongly, but independently, linked to As; Co, in addition, is closely associated with Cu, Ni, and U. Finally, V is closely associated with the elements of the main clusters: U, Cu, As, Ni, and Pb. These element associations can be interpreted in terms of ore minerals and of superimposed mineralization events (Figure 10):

(1) a first component comprises As-Ni-Co and represents Ni-Co arsenides and sulfarsenides;

(2) Cu represents another component, involving chalcopyrite;

N_t (1,2)	Ni	V	As	N_t
Ni	-	.75	1.	4
V		.50	-	6
As			-	1

7.a - samples with low tail-scores (N=7). No sample is anomalous in Cu, Zn, Pb, Ba, Sr, Co, U, or Mo.

N_t (4-7)	Cu	Zn	Pb	Ni	Ba	Sr	V	Co	As	U	Mo	N_t
Cu	-	-	.40	1.	.25	.20	.80	.80	1.	1.	-	5
Zn		-	.33	.66	.33	.66	.66	.66	1.	.66	-	3
Pb			-	1.	.50	.17	.83	.67	1.	.50	1.	6
Ni				-	.50	.57	.78	.90	1.	.73	1.	11
Ba					-	.75	.50	.50	1.	.75	.50	4
Sr						-	.43	.71	1.	.71	.50	7
V							-	.56	.78	.78	.50	9
Co								-	1.	.80	1.	10
As									-	1.	1.	14
U										-	.50	11
Mo											-	2

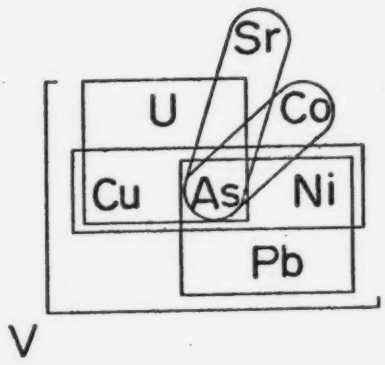
7.b - samples with medium tail-scores (N=14)

N_t (8-11)	Cu	Zn	Pb	Ni	Ba	Sr	V	Co	As	U	Mo	N_t
Cu	-	1.	1.	1.	.58	.66	1.	1.	1.	1.	.85	13
Zn		-	1.	1.	.78	.33	1.	1.	1.	1.	.77	9
Pb			-	1.	1.	1.	1.	1.	1.	1.	1.	16
Ni				-	1.	1.	1.	1.	1.	1.	1.	16
Ba					-	.66	1.	.92	1.	1.	.92	12
Sr						-	1.	.89	1.	1.	.89	9
V							-	1.	1.	1.	1.	16
Co								-	1.	1.	.93	15
As									-	1.	1.	16
U										-	1.	16
Mo											-	14

7.c - samples with high tail-scores (N=16).

Table 7. Normalized tail associations for 11 elements analyzed in clay-altered rocks from the Midwest Lake Property. Groups of samples defined in Figure 8. See Figure 9 for element associations.

9.a - group N_t (1,2)



9.b - group N_t (4-7)

Figure 9. Element associations for 11 elements analyzed in clay-altered samples from the Midwest Lake property. See Table 7 for normalized tail associations. See text for discussion.

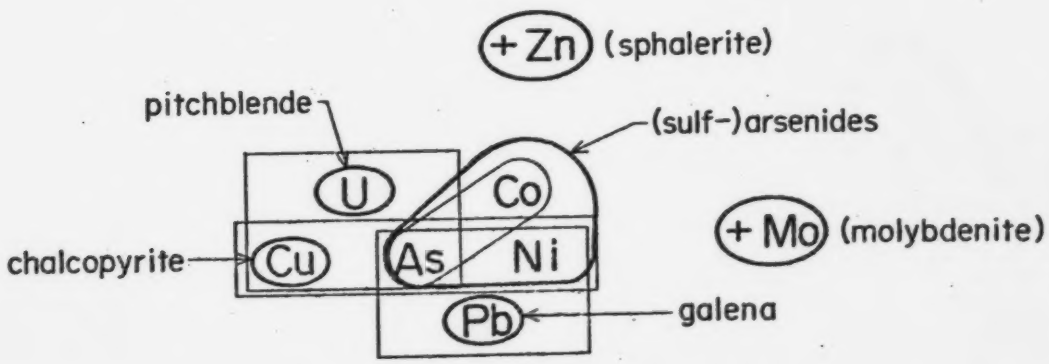


Figure 10. Interpretation of the element associations from Figure 9.b in terms of ore minerals and mineralization events. See text for discussion.

(3) U represents a pitchblende component;

(4) Pb represents a galena component.

Both the U (pitchblende) and Cu (chalcopyrite) components are closely associated to the Ni-Co-As component. Both are, however, not associated with the Pb (galena) component; the fact that Pb anomalies are not observed in the Athabasca Group in our samples (Table 6.b), may help explain this observation. Other mineralization components are: the Zn (sphalerite) component, which is antipathetic to both Cu and Pb even though all of these three components involve sulfide minerals; and finally the Mo (molybdenite) component, which is associated with both As-Ni-Co and Pb. The observation that the Mo component approaches tail saturation "late" compared to all other elements - in particular to As, Ni, Co, and Pb (Table 4) - indicates that the precipitation of molybdenite probably occurred after other ore minerals - in particular Ni-Co arsenides and sulfarsenides, and galena - had precipitated.

This example illustrates how Tails Analysis can help decipher element associations even in complex situations. In such cases, examination of the sample tail-scores will help separate different geochemical signatures and different intensities of the same signature. Of course, geological information about the samples should be used if available, in order to make the interpretation relevant or to confirm its relevancy.

III. THE TAILAN COMPUTER PROGRAM

Computer program TAILAN was written in the FORTRAN language (level 1966) to carry out Tails Analysis calculations. A description of TAILAN is given in this chapter, together with instructions for its usage (sections III.4 and following). A listing of TAILAN is given in Appendix A; a tape copy may be obtained from SRC upon request.

NOTE: the reader is referred to section 'I.4. Formalization' for details on the mathematics of Tails Analysis.

III.1 Purpose

The purpose of TAILAN is to carry out Tails Analysis calculations on an input data table of raw data. Calculations can be carried out from the raw data to the normalized tail associations table, or it can be stopped at an intermediate step. Other options include:

- calculations from a tails table;
- the saving of intermediate and final results on computer files for later use.

TAILAN carries out calculations only, up to normalized tail associations, and does not interpret results. Interpretation is left for the user to carry out.

III.2 General Structure and Input/Output

The general structure of TAILAN is shown in Figure 11. Routine TAILAN is the initial calling routine and only sets the memory vector Q(MEMORY) (see Sections III.3 and III.7). Routine CALCT is the management routine of the program. Other routines handle specific tasks. Table 8 summarizes the tasks of each routine, including input/output operations; Table 9 indicates the input/output file requirements.

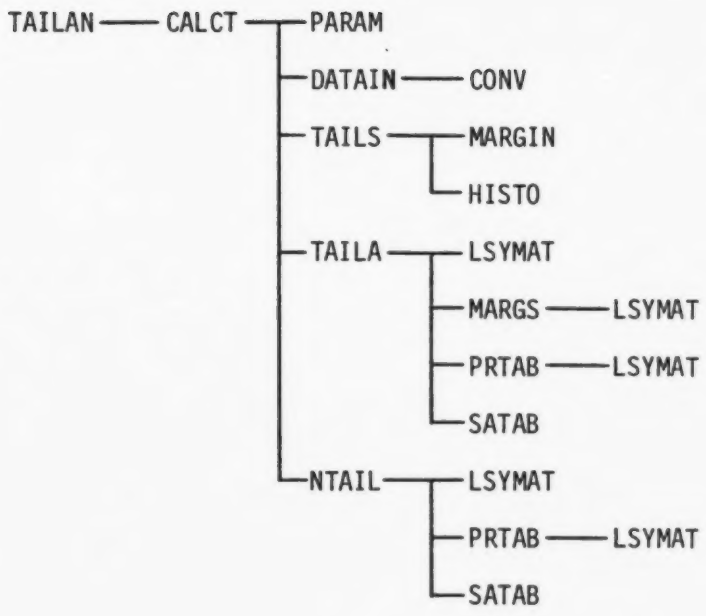


Figure 11. General Structure of TAILAN.

Routine	Tasks
TAILAN	set memory vector Q() to real size
CALCT	PRINT program header
	READ variable labels and cutoff values
	manage routine calls and memory vector Q()
PARAM	READ input parameter cards
	PRINT interpretation of input parameters, and eventual error messages
DATAIN	READ input data according to memory available
	PRINT first five rows of data table for check
TAILS	calculate tails table (T)
	SAVE (optional) variable labels, cutoff values, and tails table
	PRINT (optional) tails table
TAILA(*)	calculate tail associations (A)
NTAIL(*)	calculate normalized tail associations (AST)
	calculate average number of tail associations per sample involved (ASTK)
CONV(*)	convert special data codes prior to tails calculations
MARGIN	calculate sample tail-scores (TK)
	calculate isolated tail-scores (diag. A)
	calculate element tail-scores (TL)
HISTO	calculate histogram of sample tail-scores
	PRINT histogram of sample tail-scores
LSYMAT	calculate string location of element (i, j) of a symmetrical matrix nxn stored, as a string, in upper triangular form.
MARGS	calculate the margin of a symmetrical matrix stored, as a string, in upper triangular form.
PRTAB(*)	PRINT upper triangular matrix with two relevant margins.
SATAB(*)	SAVE variable labels, cutoff values, upper triangular matrix, margin.

(*) routine CALL is optional (see input parameters).

Table 8. Tasks of the various routines of TAILAN

I/O File	Nature	Routine Involved
Lun #15	working file (binary)	TAILS, HISTO
Lun #25	read/save file	DATAIN, TAILS
Lun #30	save file	SATAB
Lun #35	save file	SATAB

Note: Lun = Logical unit number

Table 9. Input/Output file requirements of TAILAN in addition to units #5 (Read) and #6 (Print)

COMMON name	Content
INOUT	input/output unit #'s
QDIM	dimension parameters for memory vector Q()
PRTSA	output option parameters
TITLE	title of run
DATIN	data input format, data conversion option parameter

10.a - Content of the COMMONs

(TAILAN) /	INOUT	QDIM	PRTSA	TITLE	DATIN
CALCT	X	X	X		
PARAM	X	X	X	X	X
DATAIN	X	X		X	X
TAILS	X	X	X	X	
TAILA	X	X	X		
NTAIL	X	X	X		
CONV	X	X			
MARGIN		X			
HISTO	X	X		X	
LSYMAT					
MARGS		X			
PRTAB	X	X		X	
SATAB		X			

10.b - Usage of COMMONs by the various routines.

Table 10. Content and usage of COMMONs in TAILAN

Various parameters are transferred between routines, using named COMMONs. A brief description of these COMMONs is given in Table 10.a, and their usage by routines is summarized in Table 10.b. Not transferred through COMMONs are:

- the memory vector Q(MEMORY);
- arrays containing data and/or results;
- parameters specific to some routines.

Information about these are given in the program listing.

III.3 Memory Management

Only arrays TITLE(20) and FMT(40) have a fixed dimension; all other arrays have variable dimension. This is accomplished by defining a large memory vector Q(MEMORY) in the MAIN routine:

```
DIMENSION Q(15000)
MEMORY = 15000
:
```

which will be structured to contain all required arrays. The memory vector is then transferred through a CALL statement to the program management routine CALCT and to all other routines, called by CALCT directly or indirectly. The remaining of the MAIN routine is simply:

```
:
CALL CALCT (Q,MEMORY)
STOP
END
```

Increasing the memory vector Q() is easily done (see Section III.7).

Management of the memory vector Q() is handled as follows:

Step 1: the maximum size of Q() is defined explicitly in the MAIN routine:

```
DIMENSION Q(15000)
MEMORY = 15000
```

, and is then transferred as Q(MEMORY).

Step 2: routine PARAM sets initial dimension parameters:

```
NIJ = NI*(NJ+1)
NJS = (NJ**2 + NJ)/2
```

where: NJ is the number of variables (columns) in the input data table,
NI is the number of samples (rows) in the input data table,
NIJ is the size required for storing the data table plus the sample labels,
NJS is the size required for storing the upper triangular matrix NJ x NJ.

Step 3: routine CALCT allocates successive portions of memory vector Q() to:

variable labels:
Q(1, ..., NJ)

data, including sample labels (record length NJ + 1):
Q(NJ + 1, ..., NF), NF = NJ + NIJ

cutoff values:
Q(NFM + 1, ..., MEMORY), NFM = MEMORY - NJ

If the size of Q(MEMORY) is not sufficient to contain all above items, then CALCT prints a message and aborts the run.

Step 4: routine DATAIN calculates how much room is available for sample labels and data in Q(), and reads input data until the end-of-file of the input data file is reached, or until the allocated portion of Q() is filled, whichever comes first.

Routine DATAIN then prints a message indicating how many sample records were read and whether all were read or not.

Calculations continue in any case.

Step 5: routine TAILS transforms the input data into tail codes and skips all sample records that do not have at least one tail (code = 1). The actual number of samples (rows) in the tails table is used to update the NI and NIJ dimension parameters.

Routine TAILS then verifies that memory vector Q() can accommodate the results of tails table calculations. If this test fails, then the routine prints a message and aborts the run.

If memory vector Q() is large enough, routine TAILS updates the dimension parameters of Q() for the:

data, including sample labels (record length NJ + 1):
T() = Q (NJ + 1, ..., NF), NF = NJ + NIJ

sample (row) tail-scores:
TK() = Q (NF + 1, ..., NK), NK = NJ + NI

element (column) tail-scores:
TL() = Q (NK + 1, ..., NL), NL = NK + NJ.

Step 6: routine CALCT verifies that memory vector Q() can accommodate the results of all coming calculations. If this test fails, then the routine prints a message and aborts the run.

If memory vector Q() is large enough, routine CALCT allocates new portions of Q() for the:

tail associations table:
A() = Q(NL + 1, ..., NU), NU = NL + NJS

margin of A():
AK() = Q (NU + 1, ..., NUK), NUK = NU + NJ

normalized tail associations table:
AST() = Q(NUK + 1, ..., NW), NW = NUK + NJS

average number of tail associations per sample involved:
ASTK() = Q(NW + 1, ..., NWK), NWK = NW + NJ

cutoff values:
Q(NWK + 1, ..., NX), NX = NWK + NJ

Note: routine HISTO stores the cell counts calculated from TK() back into TK(), so that initial data from this array are at least partially lost.

III.4 Input Parameters

The parameter cards required to run TAILAN, summarized in Figure 12, are read before the input data. They are:

Card #1: TITLE - Format 20A4

Card #2: General Parameters - Format 8I4

NI: number of samples (rows) in the input data. If NI=0, the program counts the samples read.

NJ: number of elements (columns) in the input data. NJ must be greater than 1.

IDATA: logical unit number on which the data are read.

If IDATA = 25, then the tails table will be read on unit #25 using the format of the file saved by the program for LST = 2 or 3 (see Section III.6).

IFMT: number of cards used to define the data input format:

= 0 assumes IDATA = 25 (no format required)
= 1 assumes one card #3
= 2 assumes two cards #3.

ICONV: data conversion option (see Section III.8):

= 0 no conversion requested.
= 1 data conversion requested prior to processing by Tails Analysis. A subroutine CONV(Q,MEMORY) must be provided, unless the default routine included in the program is suitable (see Section III.8).

LST: print/save option for tails:

= 0 the element tail-scores and a histogram of the sample tail-scores are printed.
= 1 the complete tails table is printed, along with both margins and a histogram of the sample tail-scores.
= 2 save variable labels, cutoff values, and tails table on unit #25, in addition to printing as in option 0 above.
= 3 cumulate saving above with printing as in option 1.

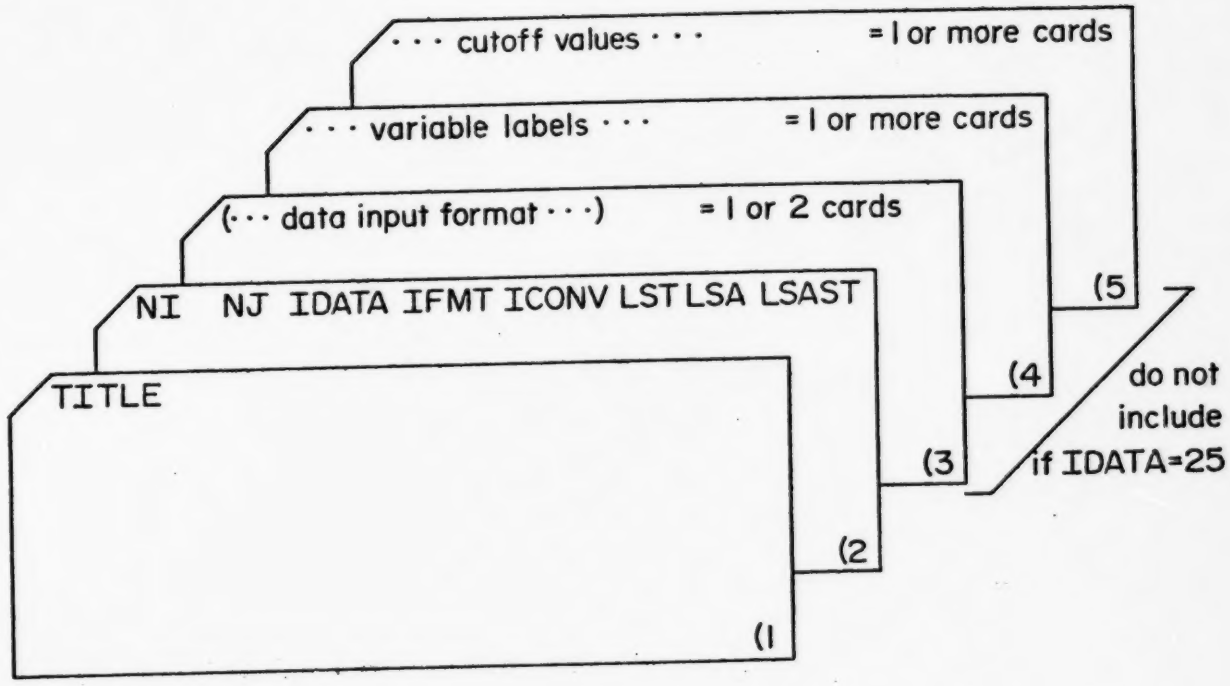


Figure 12. Input parameter cards. These are read prior to data input; see text for detailed description.

LSA: = 0 do not calculate tail associations.
= 1 print tail associations table with margins of interest.
= 2 save tail associations table with margin of interest on unit #30.
= 3 cumulate options 1 and 2 above.

LSAST: = 0 do not calculate normalized tail associations.
= 1 print normalized tail associations table with margins of interest.
= 2 save normalized tail associations table with margin of interest on unit #35.
= 3 cumulate options 1 and 2 above.

Card(s) #3: Data Input Format - Format IFMT*(20A4)
..... not required if IDATA = 25.
For each sample, the format reads:
a-the identifier, in format A;
b-the NJ element concentrations, in format F.

Card(s) #4: List of Element Names - Format 20A4
..... not required if IDATA = 25.
Note: as many cards #4 will be read as required by the value of NJ.

Card(s) #5: List of Cutoff Values - Free Format(*)
..... not required if IDATA = 25.
(*) list cutoff values in the same order as the variables listed in card(s) #4, separating two successive values by a comma.
Note: as many cards #5 will be read as required by the value of NJ.

Routine PARAM reads the input parameters and carries out various consistency checks on their values. It then prints appropriate messages before the calculations proceed. If errors are detected in the input parameters, routine PARAM will abort the run after printing error messages.

III.5 Printout

The printout for a test run is shown in Appendix B. The following are printed successively when all possible printouts are requested:

(a) contact information about Tails Analysis and TAILAN;

(b) the run title,
followed by the input parameters from card #1,
followed by an interpretation of the input parameter cards (and eventual error messages);

(c) the run title,
followed by a checklist of the first five samples as read on the input data file,
followed by a message indicating how many samples are left in the tails table after no-tail samples were eliminated during re-coding;

(d) the run title,
followed by the tails table which includes variable names, sample labels (maximum of 4 characters), tail codes, and both margins (sample and element tail-scores);
Note: a maximum of 40 samples (rows) and 20 elements (columns) are printed on one listing page. The program will automatically divide the tails table in 40 x 20 portions and print these successively on separate listing pages labelled accordingly. The portions corresponding to the same variables are printed first ("pages"), and the remaining variables are scanned in turn next ("sections").

(e) the run title,
followed by a horizontal histogram of the sample tail-scores.
Note: the number of cells is equal to NJ, the number of variables/elements, and is limited to 100 in the program (i.e., a

maximum of 100 variables/elements is assumed for the histogram). Also, up to 100 counts will be represented explicitly (by stars) on the histogram, with greater counts being represented by their printed value.

(f) the run title,

followed by comments and the tail associations table in upper triangular form. Cutoff values are also printed, as are the element tail-scores ("number of samples involved") and the margin of the tail associations table ("number of tail associations").

Note: a maximum of 40 rows and 15 columns are printed on one listing page. The program will automatically divide the tails associations table in 40 x 15 portions and print these successively on separate listing pages labelled accordingly. The portions corresponding to the same variables are scanned first ("pages"), and the remaining variables are scanned in turn next ("sections"). Only those portions which contain at least one tail association are printed.

(g) the run title,

followed by comments and the normalized tail associations table in upper triangular form. Cutoff values are also printed, as are the element tail-scores ("number of samples involved") and the average number of tail associations per sample.

Note: same print procedure as for the tail associations table (see above).

III.6 Output Files

Output files are generated by routine SATAB when appropriate values are selected for the print/save option parameters LST (tails table), LSA (tail associations table), and LSAST (normalized tail associations table). Input parameters are described in Section III.4 above.

The tails table is saved on unit #25 when LST = 2 or 3. The following are saved successively:

- the NJ variable labels (format NJ*A4),
- the NJ cutoff values (format NJ*E10.4),
- the NI rows of the tails table, each starting with the sample label (format A4, NJ*F2.0).

The tail associations table is saved on unit #30, when LSA = 2 or 3. The following are saved successively:

- the NJ variable labels (format NJ*A4),
- the NJ cutoff values (format NJ*E10.4),
- the tail associations table in upper triangular form (format NJS*F5.0),
- the element tail-scores ("number of samples involved") (format NJ*F5.0).

Notes: (a) routine LSYMAT (see Appendix A) can be used to find the location of items from the tail associations table stored in its upper triangular form as a string of values;

(b) the margin of the table can be easily calculated from the stored values using routine MARGS (see Appendix A).

The normalized tail associations table is saved on unit #35, when LSAST = 2 or 3. The following are saved successively:

- the NJ variable labels (format NJ*A4),
- the NJ cutoff values (format NJ*E10.4),
- the normalized tail associations table in upper triangular form (format NJS*F5.0),
- the element tail-scores ("number of samples involved") (format NJ*F5.0).

Notes: (a) routine LSYMAT (see Appendix A) can be used to find the location of items from the tail associations table stored in its upper triangular from as a string of values;

(b) all normalized tail associations are multiplied by 100 before storage;

(c) the average number of tail associations per sample involved can be calculated using the normalized tail associations and the element tail-scores, both saved on file, using the procedure described in Section I.4 (last paragraph). Routines LSYMAT and MARGS (see Appendix A) will be useful for these calculations.

III.7 Increasing the Memory Vector

Memory management in TAILAN, using memory vector Q(), is described in Section III.3 above. In order to increase the size of memory vector Q(), in which all arrays are stored in variable dimension, only the first two statements of the MAIN routine need to be modified:

```
default:            DIMENSION Q(15000)
                    MEMORY = 15000

modified (e.g.):    DIMENSION Q(30000)
                    MEMORY = 30000
```

If TAILAN is stored as a load module (object program), then the MAIN routine must be recompiled and the Linkage Editor invoked to override the corresponding control section of the load module.

III.8 Modifying the Data Conversion Routine

Data conversion prior to processing by routine TAILS may be required and is accomplished by routine CONV, called by DATAIN if ICONV=1 on input parameter card #1 (see Section III.4). Data conversion involves the translation of special concentration codes into actual concentration values, so that tails are calculated properly for all samples.

The data conversion routine included in TAILAN treats special codes from SRC's Lithogeochemistry Data File (Mellinger, 1983) - see routine CONV in Appendix A.

If another data conversion scheme is required, the user must provide a new CONV routine similar in structure to the one provided here. If TAILAN is stored as a load module (object program), then the new CONV routine must be compiled and the Linkage Editor invoked to override the corresponding control section of the load module.

Note: routine CONV may be used to perform relevant data conversions other than special code conversion, before actual processing by Tails Analysis begins.

REFERENCES

Mellinger, M. (1981). Geology and Geochemistry of the Maurice Bay uranium deposit (Sask.): Final Report. SRC Publication No. G-740-3-C-81 (Confidential).

Mellinger, M. (1982). The Maurice Bay uranium deposit (Sask.): geology, host-rock alteration and genesis. SRC Publication No. G-740-8-D-82. Accepted for publication in the CIM Uranium Symposium Volume (Saskatchewan 1981), CIM (1984).

Mellinger, M., Hoeve, J. (1982). Geochemical and mineralogical investigation of diabase and "mudstone" occurrences - Midwest Lake property (Sask.). SRC Publication No. G-740-6-C-82.

Mellinger, M. (1983). The Lithogeochemistry Data File: user manual. SRC Publication No. R-740-1-E-83.

APPENDIX A

Listing of the TAILAN Computer Program

Notes: - TAILAN is written in FORTRAN language, version 1966;
- routines are listed here in alphabetical order;
- information about TAILAN is given in Chapter III of this report.
- a tape copy of TAILAN is available on request.

```
C*****************************************************************************00000010
C                                                                              00000020
C     TAILAN - TAILS ANALYSIS PROGRAM.                                         00000030
C                                                                              00000040
C        WRITTEN BY MICHEL MELLINGER                                           00000050
C                   RESOURCES SECTOR                                           00000060
C                   SASKATCHEWAN RESEARCH COUNCIL                              00000070
C                   30, CAMPUS DRIVE,                                          00000080
C                   SASKATOON (SASK.)                                          00000090
C                   CANADA - S7N 0X1                                           00000100
C                                                                              00000110
C        PLEASE CONSULT SRC PUBLICATION # R-740-4-B-83,                        00000120
C           FOR COMPLETE INSTRUCTIONS ABOUT THE METHOD AND PROGRAM.            00000130
C                                                                              00000140
C     TAILAN---CALCT---PARAM                                                   00000150
C                   +--DATAIN---CONV                                           00000160
C                   +--TAILS---MARGIN                                          00000170
C                            +--HISTO                                          00000180
C                   +--TAILA---LSYMAT                                          00000190
C                            +--MARGS---LSYMAT                                 00000200
C                            +--PRTAB---LSYMAT                                 00000210
C                            +--SATAB                                          00000220
C                   +--NTAIL---LSYMAT                                          00000230
C                            +--PRTAB---LSYMAT                                 00000240
C                            +--SATAB                                          00000250
C                                                                              00000260
C MINIMUM MEMORY REQUIRED = NI*(NJ+1) + NJ**2 + 6*NJ + NI  ,                   00000270
C    WHERE: NI=NUMBER OF ROWS IN THE DATA TABLE (INDIVIDUALS)                  00000280
C           NJ=NUMBER OF COLUMNS IN THE DATA TABLE (VARIABLES)                 00000290
C                                                                              00000300
C*****************************************************************************00000310
C OCT 6/82: M.M.(SRC) RELEASE 1 - FINAL TESTS COMPLETED JUNE 25,1983.          00000320
C UPDATES :                                                                    00000330
C*****************************************************************************00000340
      DIMENSION Q(15000)                                                       00000350
      MEMORY = 15000                                                           00000360
      CALL CALCT(Q,MEMORY)                                                     00000370
      STOP                                                                     00000380
      END                                                                      00000390
```

```
                                                                      CALCT/
      SUBROUTINE CALCT(G,MEMORY)                                      00000400
C*****************************************************************************00000410
C  SUBROUTINE CALCT IS THE MANAGEMENT ROUTINE FOR PROGRAM TAILAN       00000420
C                                                                      00000430
C     CALLED BY TAILAN (MAIN)                                          00000440
C     CALLS PARAM                                                      00000450
C           DATAIN...                                                  00000460
C           TAILS ...                                                  00000470
C           TAILA ...                                                  00000480
C           NTAIL ...                                                  00000490
C                                                                      00000500
C     TASKS: PRINT PROGRAM HEADER                                      00000510
C            MANAGE VECTOR G() AT VARIOUS STEPS                        00000520
C            CHECK MEMORY REQUIREMENTS AT VARIOUS STEPS                00000530
C            CALL PARAMETER INPUT ROUTINE                              00000540
C            READ LABEL() AND OUTSF()                                  00000550
C            CALL DATA INPUT AND CALCULATION ROUTINES                  00000560
C                                                                      00000570
C     I/O UNITS : IDATA = DATA INPUT UNIT, DEFINED IN PARAMETER CARDS  00000580
C                 IFT,IFA,IFAST = DATA OUTPUT UNITS FOR THE TAILS TABLE,00000590
C                    THE TAIL ASSOCIATIONS TABLE, AND THE NORMALIZED   00000600
C                    TAIL ASSOCIATIONS TABLE, RESPECTIVELY/ DEFINED AS 00000610
C                    LUN = 25,30,35/ USER SHALL PRODUCE CORRECT JCL    00000620
C                 IFTEMP = WORKING FILE USED FOR TEMPORARY BINARY      00000630
C                    STORAGE/ DEFINED AS LUN = 15...SEE JCL            00000640
C                                                                      00000650
C MINIMUM MEMORY REQUIRED = NI*(NJ+1) + NJ**2 + 6*NJ + NI ,            00000660
C    WHERE: NI=NUMBER OF ROWS IN THE DATA TABLE (INDIVIDUALS)          00000670
C           NJ=NUMBER OF COLUMNS IN THE DATA TABLE (VARIABLES)         00000680
C                                                                      00000690
C OTHER MAIN NOTATIONS:                                                00000700
C    NJ1 = NJ + 1                                                     00000710
C    NIJ = NI*(NJ+1) = SIZE OF THE DATA TABLE, INCLUDING SAMPLE LABELS 00000720
C    NJS = (NJ**2 + NJ)/2 = SIZE OF THE UPPER TRIANGULAR MATRIX NJ*NJ  00000730
C    GENERAL PARAMETERS : SEE INSTRUCTIONS, PARAMETER CARD#2           00000740
C    VECTOR Q() PARAMETERS : SEE SRC REPORT R-740-4-B-83               00000750
C                                                                      00000760
C*****************************************************************************00000770
C MICHEL MELLINGER (SRC), OCT 6/82. FINAL TESTS COMPLETED JUNE25,1983. 00000780
C UPDATES :                                                            00000790
C*****************************************************************************00000800
      COMMON/INOUT/IREAD,IPRT,IDATA,IFT,IFA,IFAST,IFTEMP               00000810
      COMMON/QDIM/NI,NJ,NJ1,NIJ,NJS                                    00000820
      COMMON/PRTSA/LST,LSA,LSAST                                       00000830
      DIMENSION Q(MEMORY)                                              00000840
C                                                                      00000850
C SET UP LOGICAL UNIT NUMBERS ('IF---' FOR MASS STORAGE UNITS)         00000860
      IREAD = 5                                                        00000870
      IPRT = 6                                                         00000880
      IFTEMP = 15                                                      00000890
      IFT    = 25                                                      00000900
      IFA    = 30                                                      00000910
      IFAST  = 35                                                      00000920
C                                                                      00000930
C PRINT PROGRAM HEADER                                                 00000940
      WRITE(IPRT,1000)                                                 00000950
      WRITE(IPRT,1005)                                                 00000960
C                                                                      00000970
C READ AND SET UP PROGRAM PARAMETERS, QUIT IF ERROR(S) ENCOUNTERED     00000980
      NERROR = 0                                                       00000990
      CALL PARAM(NERROR)                                               00001000
      IF(NERROR.NE.0) RETURN                                           00001010
```

```
C                                                                       00001020
C SET UP VECTOR Q(). PRELIMINARY PARAMETER SET-UP WAS MADE              00001030
C  IN ROUTINE PARAM                                                    00001040
      NF = NJ + NIJ                                                     00001050
      NFJ = NF + NJ                                                     00001060
      NFM = MEMORY - NJ                                                 00001070
      NFM1 = NFM + 1                                                    00001080
C                                                                       00001090
C FIRST MEMORY CHECK                                                    00001100
      IF(NFJ.GT.MEMORY) WRITE(IPRT,1010)MEMORY,NFJ                      00001110
      IF(NFJ.GT.MEMORY) RETURN                                          00001120
C                                                                       00001130
C IF IDATA = 25, THE INPUT DATA FILE IS A TAILS TABLE,                  00001140
C  SKIP READING LABELS AND CUTOFFS ON UNIT IREAD                        00001150
      IF(IDATA.EQ.25) GO TO 10                                          00001160
C                                                                       00001170
C READ VARIABLE LABELS(IN A4) AND CUTOFF VALUES(FREE FORMAT)            00001180
C STORE RESPECTIVELY AT BEGINNING AND END OF Q()                        00001190
         READ(IREAD,1500) (Q(K),K=1,NJ)                                 00001200
         READ(IREAD,*) (Q(K),K=NFM1,MEMORY)                             00001210
   10 CONTINUE                                                          00001220
C                                                                       00001230
C READ INPUT DATA                                                       00001240
      CALL DATAIN(Q,MEMORY)                                             00001250
C                                                                       00001260
C CALCULATE TAILS                                                       00001270
      CALL TAILS(Q,MEMORY,Q(NFM+1))                                     00001280
C                                                                       00001290
C IF MEMORY CHECK IN TAILS WAS NEGATIVE, OR                             00001300
C  IF NONE OF THE SAMPLES HAS A TAIL.....END PROGRAM                    00001310
      IF(MEMORY.EQ.0) RETURN                                            00001320
C                                                                       00001330
C FINAL MEMORY CHECK BEFORE PROCEEDING, QUIT IF CHECK NEGATIVE          00001340
      MEMCHK = NIJ + (NJ**2) + (6*NJ) + NI                              00001350
      IF(MEMORY.GE.MEMCHK) GO TO 50                                     00001360
         WRITE(IPRT,1015) MEMORY,MEMCHK                                 00001370
         RETURN                                                         00001380
   50 CONTINUE                                                          00001390
C                                                                       00001400
C RESET VECTOR Q() PARAMETERS TO FINAL VALUES                           00001410
C     NF  = NJ + NIJ                                                    00001420
C     NK  = NF + NI      | ...SET IN ROUTINE TAILS                     00001430
C     NL  = NK + NJ      /                                            00001440
      NU  = NL + NJS                                                    00001450
      NUK = NU + NJ                                                     00001460
      NW  = NUK + NJS                                                   00001470
      NWK = NW  + NJ                                                    00001480
      NX  = NWK + NJ                                                    00001490
C                                                                       00001500
C RELOCATE CUTOFF VALUES AT END OF VECTOR Q()                           00001510
      DO 100 J=1,NJ                                                     00001520
         NCA = NWK + J                                                  00001530
         NCB = NFM + J                                                  00001540
         Q(NCA) = Q(NCB)                                                00001550
  100 CONTINUE                                                          00001560
C                                                                       00001570
C CALCULATE TAIL ASSOCIATIONS                                           00001580
      IF(ILSA.NE.0) CALL TAILA(Q(1),Q(NJ+1),Q(NK+1),                    00001590
     $                         Q(NL+1),Q(NU+1),Q(NWK+1))                00001600
C                                                                       00001610
C CALCULATE NORMALIZED TAIL ASSOCIATIONS                                00001620
      IF(ILSAST.NE.0) CALL NTAIL(Q(1),Q(NK+1),Q(NL+1),Q(NU+1),          00001630
     $                           Q(NUK+1),Q(NW+1),Q(NWK+1))             00001640
```

```
C                                                                       00001650
C END OF RUN                                                            00001660
      WRITE(IPRT,1020)                                                  00001670
C                                                                       00001680
C FORMATS                                                               00001690
 1000 FORMAT(1H1///10X,46X,15H|  *  (0)  | (1)//                        00001700
     1 10X,46X,             11H|  **     |/                            00001710
     2 10X,46X,             11H| *****   //                            00001720
     3 10X,8H********,38X, 15H| ********  */                           00001730
     4 10X,41H*TAILAN*      TAILS ANALYSIS PROGRAM (SRC),               00001740
     5 5X,18H|**************  */                                        00001750
     6 10X,8H********,38X,29H+******************** *  *   *///         00001760
     7 10X,78HRELEASE 1 (V.1)...INFORMATION ABOUT TAILS ANALYSIS AND IN00001770
     8STRUCTIONS FOR TAILAN/28X,49H CAN BE FOUND IN SRC PUBLICATION # R-00001780
     940-4-8-83.   )                                                    00001790
 1005 FORMAT(33X,38HCONTACT: SASKATCHEWAN RESEARCH COUNCIL/             00001800
     1       33X,38H           30, CAMPUS DRIVE          /              00001810
     2       33X,38H           SASKATOON (SASK.)         /              00001820
     3       33X,38H           CANADA - S7N 0A1          /              00001830
     4       33X,38H         (PH.: 306/664-5400)         )              00001840
 1010 FORMAT(1H1/////10X,44HCALCT-INSUFFICIENT MEMORY SPACE (FIRST TEST)00001850
     1//10X,13HYOU RESERVED ,I10,12H...YOU NEED ,I10,33H AT THIS STAGE (00001860
     2SEE INSTRUCTIONS))                                                00001870
 1015 FORMAT(1H1/////10X,44HCALCT-INSUFFICIENT MEMORY SPACE (FIRST TEST)00001880
     1//10X,13HYOU RESERVED ,I10,12H...YOU NEED ,I10,19H (SEE INSTRUCTI00001890
     2NS)  )                                                            00001900
 1020 FORMAT(1H1/////50X,27H*****NORMAL END OF JOB*****)                00001910
 1500 FORMAT(20A4)                                                      00001920
C                                                                       00001930
      RETURN                                                            00001940
      END                                                               00001950
```

```
      SUBROUTINE PARAM(NERROR)                                          00001960
C*********************************************************************00001970
C SUBROUTINE PARAM: READS IN AND CHECKS INPUT PARAMETER CARD            00001980
C                   SETS UP ADDITIONAL PROGRAM PARAMETERS               00001990
C                                                                       00002000
C     NERROR IS THE NUMBER OF ERRORS DETECTED IN THE INPUT CARDS        00002010
C                                                                       00002020
C     CALLED BY CALCT                                                   00002030
C                                                                       00002040
C     TASKS : READ TITLE                                                00002050
C             READ PARAMETER CARD, CHECK PARAMETERS FOR VALIDITY        00002060
C             READ DATA INPUT FORMAT                                    00002070
C             SET UP INITIAL DIMENSION PARAMETERS OF VECTOR NI          00002080
C                                                                       00002090
C*********************************************************************00002100
C MICHEL MELLINGER (SRC), OCT 6/82                                      00002110
C UPDATES :                                                             00002120
C*********************************************************************00002130
      COMMON/INOUT/IREAD,IPRT,IDATA,IFT,IFA,IFAST,IFTEMP                00002140
      COMMON/TITLE/TITLE                                                00002150
      COMMON/QDIM/NI,NJ,NJ1,NIJ,NJS                                     00002160
      COMMON/DATIN/FMT,ICONV                                            00002170
      COMMON/PRTSA/LST,LSA,LSAST                                        00002180
      DIMENSION TITLE(20),FMT(40)                                       00002190
C                                                                       00002200
C READ TITLE, PARAMETERS, AND DATA INPUT FORMAT                         00002210
      READ(IREAD,1000) TITLE                                            00002220
      READ(IREAD,1010) NI,NJ,IDATA,IFMT,ICONV,LST,LSA,LSAST             00002230
C                                                                       00002240
C PRINT TITLE OF RUN                                                    00002250
      WRITE(IPRT,1500) TITLE                                            00002260
C                                                                       00002270
C PRINT GENERAL PARAMETERS FOR CHECK                                    00002280
      WRITE(IPRT,1505) NI,NJ,IDATA,IFMT,ICONV,LST,LSA,LSAST             00002290
C                                                                       00002300
C CHECK PARAMETERS AND PRINT ERROR MESSAGES IF ERROR(S) FOUND           00002310
C NERROR = NUMBER OF ERRORS ENCOUNTERED (SET AT 0 IN CALCT BEFORE       00002320
C  THE SUBROUTINE IS CALLED)                                            00002330
      WRITE(IPRT,1670)                                                  00002340
C                                                                       00002350
C FIRST COUNT NUMBER OF ERRORS AND PRINT DIAGNOSTIC                     00002360
      IF(NI.LT.0) NERROR = NERROR + 1                                   00002370
      IF(NJ.LT.2) NERROR = NERROR + 1                                   00002380
      IF((IFMT.LE.0.AND.IDATA.NE.25) NERROR = NERROR + 1                00002390
      IF(IFMT.GT.2) NERROR = NERROR + 1                                 00002400
      IF(LST.LT.0.OR.LSA.LT.0.OR.LSAST.LT.0) NERROR = NERROR + 1        00002410
      IF(LST.GT.3.OR.LSA.GT.3.OR.LSAST.GT.3) NERROR = NERROR + 1        00002420
      IF(LSAST.GE.1.AND.LSA.EQ.0) NERROR = NERROR + 1                   00002430
         IF(NERROR.EQ.0) WRITE(IPRT,1680)                               00002440
         IF(NERROR.NE.0) WRITE(IPRT,1690)NERROR                         00002450
C                                                                       00002460
C     LIST ERROR(S) FOUND IF ANY                                        00002470
      IF(NERROR.EQ.0) GO TO 100                                         00002480
```

```
      IF(NI.LT.0) WRITE(IPRT,1700)
      IF(NJ.LT.2) WRITE(IPRT,1710)
      IF(IFMT.LE.0.AND.IDATA.NE.25) WRITE(IPRT,1720)
      IF(IFMT.GT.2) WRITE(IPRT,1730)
      IF(LST.LT.0.OR.LSA.LT.0.OR.LSAST.LT.0) WRITE(IPRT,1740)
      IF(LST.GT.3.OR.LSA.GT.3.OR.LSAST.GT.3) WRITE(IPRT,1750)
      IF(LSAST.GE.1.AND.LSA.EQ.0) WRITE(IPRT,1755)
C
C     END RUN IF ERROR(S) ENCOUNTERED
      RETURN
  100 CONTINUE
C
C PRINT GENERAL PARAMETERS, WITH EVENTUAL MESSAGES
      IF(NI.EQ.0) WRITE(IPRT,1510)NJ
      IF(NI.GT.0) WRITE(IPRT,1520)NI,NJ
      IF(IDATA.EQ.25) WRITE(IPRT,1530)
      IF(IDATA.NE.25) WRITE(IPRT,1540)IDATA
      IF(ICONV.NE.0) WRITE(IPRT,1541)
      WRITE(IPRT,1545) LST,LSA,LSAST
      ISAVE = LST + 1
      GO TO(10,11,12,13),ISAVE
   10    WRITE(IPRT,1550)
         GO TO 40
   11    WRITE(IPRT,1560)
         GO TO 40
   12    WRITE(IPRT,1550)
         IF(IDATA.NE.25) WRITE(IPRT,1570)
         GO TO 40
   13    WRITE(IPRT,1560)
         IF(IDATA.NE.25) WRITE(IPRT,1570)
   40 CONTINUE
      ISAVE = LSA + 1
      GO TO(20,21,22,23),ISAVE
   20    WRITE(IPRT,1590)
         GO TO 50
   21    WRITE(IPRT,1600)
         GO TO 50
   22    WRITE(IPRT,1590)
         WRITE(IPRT,1610)
         GO TO 50
   23    WRITE(IPRT,1600)
         WRITE(IPRT,1610)
   50 CONTINUE
      ISAVE = LSAST + 1
      GO TO(30,31,32,33),ISAVE
   30    WRITE(IPRT,1630)
         GO TO 60
   31    WRITE(IPRT,1640)
         GO TO 60
   32    WRITE(IPRT,1630)
         WRITE(IPRT,1610)
         GO TO 60
   33    WRITE(IPRT,1640)
         WRITE(IPRT,1610)
   60 CONTINUE
C
C READ AND PRINT DATA INPUT FORMAT
      IF(IDATA.EQ.25) GO TO 110
      READ(IREAD,1000) (FMT(K),K=1,20)
      IF(IFMT.EQ.2) READ(IREAD,1000)(FMT(K),K=21,40)
      WRITE(IPRT,1700) IFMT,FMT
  110 CONTINUE
```

```
C                                                                       00003110
C SET UP INITIAL DIMENSION PARAMETERS FOR VECTOR NI                     00003120
      NJ1 = NJ + 1                                                      00003130
      NIJ = NI*(NJ+1)                                                   00003140
      NJS = (NJ**2 + NJ)/2                                              00003150
C                                                                       00003160
C FORMATS                                                               00003170
 1000 FORMAT(20A4)                                                      00003180
 1010 FORMAT(3I4)                                                       00003190
 1500 FORMAT(1H1///6X,16HTITLE OF RUN IS ,20A4///)                      00003200
 1505 FORMAT(6X,56H     NI     NJ  IDATA   IFMT  ICONV    LST    LSA  LS00003210
     $AST/ 6X,8I7///)                                                   00003220
 1510 FORMAT(6X,61HTHE NUMBER OF SAMPLES WILL BE COUNTED, NUMBER OF VARI00003230
     $ABLES = ,I3)                                                      00003240
 1520 FORMAT(6X,16HINPUT TABLE HAS ,I8,11H SAMPLES X ,I8,10H VARIABLES) 00003250
 1530 FORMAT(6X,20HINPUT DATA UNIT   25,23H (TAILS TABLE AS INPUT))     00003260
 1540 FORMAT(6X,16HINPUT DATA UNIT ,I4,20H (RAW DATA AS INPUT))         00003270
 1541 FORMAT(10X,31HSPECIAL CODES WILL BE CONVERTED)                    00003280
 1545 FORMAT(6X,23HOUTPUT OPTIONS : LST = ,I2,3H ,LSA = ,I2,            00003290
     $10H ,LSAST = ,I2)                                                 00003300
 1550 FORMAT(10X,68HTAILS TABLE : PRINT BOTTOM MARGIN AND HISTOGRAM OF R00003310
     $IGHT MARGIN ONLY)                                                 00003320
 1560 FORMAT(10X,71HTAILS TABLE : PRINT TABLE WITH MARGINS, PRINT HISTOG00003330
     $RAM OF RIGHT MARGIN)                                              00003340
 1570 FORMAT(24X,16HSAVE TAILS TABLE)                                   00003350
 1590 FORMAT(10X,36HTAIL ASSOCIATIONS : DO NOT PRINT TABLE)             00003360
 1600 FORMAT(10X,58HTAIL ASSOCIATIONS : PRINT TABLE AND QUANTITIES OF IN00003370
     $TEREST)                                                           00003380
 1610 FORMAT(30X,37HSAVE TABLE AND QUANTITIES OF INTEREST)              00003390
 1630 FORMAT(10X,38HNORMALIZED  "  "  : DO NOT PRINT TABLE)             00003400
 1640 FORMAT(10X,58HNORMALIZED  "  "  : PRINT TABLE AND QUANTITIES OF IN00003410
     $TEREST)                                                           00003420
 1670 FORMAT(1HO,5X,18HPARAMETER CHECK : )                              00003430
 1680 FORMAT(1H+,23X,2HOK)                                              00003440
 1690 FORMAT(1H+,23X,I2,15H ERROR(S) FOUND)                             00003450
 1700 FORMAT(10X,34H...NEGATIVE NUMBER OF SAMPLES (NI))                 00003460
 1710 FORMAT(10X,28H...NOT ENOUGH VARIABLES (NJ))                       00003470
 1720 FORMAT(10X,89H...DATA INPUT FORMAT CARDS NEEDED (IFMT), OR SET IDA00003480
     $TA TO 25 TO READ TAILS TABLE AS INPUT)                            00003490
 1730 FORMAT(10X,54H...MAXIMUM OF 2 DATA INPUT FORMAT CARDS ALLOWED (IFM00003500
     $T))                                                               00003510
 1740 FORMAT(10X,66H...ONE AT LEAST OF THE OUTPUT PARAMETERS IS NEGATIVE00003520
     $ (LST,LSA,LSAST))                                                 00003530
 1750 FORMAT(10X,74H...ONE AT LEAST OF THE OUTPUT PARAMETERS IS GREATER 00003540
     $THAN 3 (LST,LSA,LSAST))                                           00003550
 1755 FORMAT(10X,75H...NORMALIZED TAIL ASSOCIATIONS CANNOT BE CALCULATED00003560
     $: SET LSA TO 1 OR MORE  )                                         00003570
 1760 FORMAT(6X,19HDATA INPUT FORMAT (,I2,12H CARD(S)) : ,20A4/,9X,20A4)00003580
C                                                                       00003590
      RETURN                                                            00003600
      END                                                               00003610
```

```
      SUBROUTINE DATAIN(Q,MEMORY)                                       00003620
C*************************************************************************00003630
C SUBROUTINE DATAIN READS IN THE DATA TABLE (RAW DATA OR TAILS)          00003640
C                                                                        00003650
C     CALLED BY CALCT                                                    00003660
C     CALLS CONV (IF ICONV.NE.0)                                         00003670
C                                                                        00003680
C     TASKS : READ IN DATA TABLE (OR TAILS TABLE)                        00003690
C                COUNT NUMBER OF SAMPLES IF NOT GIVEN IN INPUT           00003700
C             PRINT FIRST FIVE ROWS OF DATA TABLE (OR TAILS TABLE)       00003710
C                WITH: HEADER FOR VARIABLES,                             00003720
C                      SET OF CORRESPONDING CUTOFF VALUES, AND           00003730
C                      NUMBER OF SAMPLES (TOTAL NUMBER OF ROWS)          00003740
C             CHECK SIZE OF VECTOR Q() FOR STORING DATA, AND READ AND    00003750
C                STORE DATA ACCORDING TO AVAILABLE SPACE                 00003760
C                                                                        00003770
C*************************************************************************00003780
C MICHEL MELLINGER (SRC), OCT 6/82.                                      00003790
C UPDATES :                                                              00003800
C*************************************************************************00003810
      COMMON/TITLE/TITLE                                                 00003820
      COMMON/INOUT/IREAD,IPRT,IDATA,IPT,IFA,IFAST,IFTEMP                 00003830
      COMMON/DIM/NI,NJ,NJI,NIJ,NJS                                       00003840
      COMMON/DATIN/FMT,ICONV                                             00003850
      DIMENSION Q(MEMORY)                                                00003860
      DIMENSION TITLE(20),FMT(40)                                        00003870
C                                                                        00003880
C READ INPUT DATA (RAW DATA, OR TAILS TABLE)                             00003890
C                                                                        00003900
C     FIRST CALCULATE HOW MUCH SPACE IS LEFT FOR DATA IN VECTOR Q(),     00003910
C      AND READ EITHER NI OR NSAMP<=NI SAMPLES                           00003920
      NSAMP = (MEMORY-(2*NJ)) / (NJ+1)                                   00003930
      IF(NI.GT.0) NSAMP = NI                                             00003940
C                                                                        00003950
C     COUNT ACTUAL NUMBER OF SAMPLES READ                                00003960
C      INITIALIZE COUNTER AND FIRST MEMORY LOCATION OF CUTOFF VALUES     00003970
      NCOUNT = 0                                                         00003980
      NFM1 = (MEMORY - NJ) + 1                                           00003990
C                                                                        00004000
C     IF IDATA = 25, READ DIRECTLY TAILS TABLE                           00004010
      IF(IDATA.EQ.25) GO TO 50                                           00004020
C                                                                        00004030
C     ...READ RAW DATA                                                   00004040
         DO 10 K=1,NSAMP                                                 00004050
            LL1 = K*(NJ + 1)                                             00004060
            LL2 = LL1 + NJ                                               00004070
            READ(IDATA,FMT,END=20) (Q(L),L=LL1,LL2)                      00004080
            NCOUNT = NCOUNT + 1                                          00004090
   10    CONTINUE                                                        00004100
   20    CONTINUE                                                        00004110
         GO TO 100                                                       00004120
      50 CONTINUE                                                        00004130
C                                                                        00004140
C     ...NO RAW DATA: READ VARIABLE LABELS, CUTOFFS, THEN TAILS TABLE    00004150
         READ(25,1000) (Q(L),L=1,NJ)                                     00004160
         READ(25,1010) (Q(L),L=NFM1,MEMORY)                              00004170
```

```
      DO 30 K=1,NSAMP                                                   00004180
         LL1 = K*(NJ + 1)                                               00004190
         LL2 = LL1 + NJ                                                 00004200
         READ(25,900,END=40) (Q(L),L=LL1,LL2)                           00004210
         NCOUNT = NCOUNT + 1                                            00004220
   30    CONTINUE                                                       00004230
   40    CONTINUE                                                       00004240
  100 CONTINUE                                                          00004250
C                                                                       00004260
C END OF DATA INPUT                                                     00004270
C VERIFY IF END OF FILE WAS REACHED, CALCULATIONS CONTINUE IN ANY CASE  00004280
C                                                                       00004290
C     PRINT PAGE HEADER                                                 00004300
      WRITE(IPRT,1490) TITLE                                            00004310
C                                                                       00004320
C     IF NCOUNT = NSAMP, END OF FILE WAS NOT REACHED (NOT ALL           00004330
C      SAMPLES WERE READ)                                               00004340
        IF(NI.EQ.0.AND.NCOUNT.EQ.NSAMP) WRITE(IPRT,1500)NCOUNT          00004350
        IF(NI.EQ.0.AND.NCOUNT.LT.NSAMP)  WRITE(IPRT,1510)NCOUNT         00004360
        IF(NI.GT.0.AND.NCOUNT.LT.NI) WRITE(IPRT,1520)NCOUNT,NI,IDATA    00004370
C                                                                       00004380
C RESET PARAMETERS OF VECTOR Q() ACCORDING TO NEW NI VALUE              00004390
      NI = NCOUNT                                                       00004400
      NIJ = NI*(NJ + 1)                                                 00004410
C                                                                       00004420
C DATA CHECK PRINTING :                                                 00004430
C PRINT FIRST FIVE SAMPLES READ,PRECEDED BY HEADER VARIABLE LABELS AND  00004440
C  CUTOFF VALUES                                                        00004450
        IF(IDATA.NE.25) WRITE(IPRT,1530)                                00004460
        IF(IDATA.EQ.25) WRITE(IPRT,1570)                                00004470
        WRITE(IPRT,1540) (Q(L),L=1,NJ)                                  00004480
        WRITE(IPRT,1550) (Q(L),L=NFMI,MEMORY)                           00004490
        DO 110 K=1,5                                                    00004500
           LL1 = K*(NJ + 1)                                             00004510
           LL2 = LL1 + NJ                                               00004520
           WRITE(IPRT,1590) (Q(L),L=LL1,LL2)                            00004530
  110   CONTINUE                                                        00004540
C                                                                       00004550
C CONVERT SPECIAL CODES IN DATA (CUSTOMIZED ROUTINE)                    00004560
      IF(ICONV.NE.0) CALL CONV(Q,MEMORY)                                00004570
C                                                                       00004580
C FORMATS                                                               00004590
  900 FORMAT(A4,38F2.0/(40F2.0))                                        00004600
 1000 FORMAT(20A4/(20A4))                                               00004610
 1010 FORMAT(8E10.4/(8E10.4))                                           00004620
 1490 FORMAT(1H1///6X,17HTAILS ANALYSIS : ,20A4)                        00004630
 1500 FORMAT(1HO,6X,I3,65H SAMPLES READ.....END OF FILE NOT REACHED...CA00004640
     . $LCULATIONS CONTINUE///)                                         00004650
 1510 FORMAT(1HO,6X,I3,56H SAMPLES READ.....END OF FILE REACHED (ALL SA00004660
     $PLES READ)///)                                                    00004670
 1520 FORMAT(1HO,6X,5HONLY ,I3,21H SAMPLES READ OF THE ,I3,49H ASKED BEF00004680
     $, DUE TO END OF FILE ON INPUT UNIT ,I4///)                        00004690
 1530 FORMAT(1HO,5X,33HFIRST FIVE ROWS OF DATA AS READ :)               00004700
 1540 FORMAT(1HO,5X,8HVARIABLE,10(4X,A4,3X)/(14X,10(4X,A4,3X)))         00004710
 1550 FORMAT(6X,6H(CUTOFF),10(1X,G10.4)/(14X,10(1X,E10.4)))             00004720
 1570 FORMAT(5X,39HFIRST FIVE ROWS OF TAILS TABLE AS READ:)             00004730
 1590 FORMAT(1HO,7X,A4,2X,10(F10.2,1X)/(14X,10(F10.2,1X)))              00004740
C                                                                       00004750
      RETURN                                                            00004760
      END                                                               00004770
```

```
      SUBROUTINE TAILS(Q,MEMORY,CUTOF)                                  00004780
C*****************************************************************     00004790
C SUBROUTINE TAILS CALCULATES THE TAILS TABLE AND RELATED INFORMATION.  00004800
C                                                                       00004810
C     NOTE: THE TAIL CODES ARE STORED IN Q() OVERRIDING THE INITIAL     00004820
C           DATA (I.E. THE INITIAL DATA ARE LOST).                      00004830
C                                                                       00004840
C     CUTOF() CONTAINS THE CUTOFF VALUES USED TO DEFINE THE TAILS       00004850
C                                                                       00004860
C     CALLED BY CALCT                                                   00004870
C     CALLS MARGIN                                                      00004880
C           HISTO...                                                    00004890
C                                                                       00004900
C     TASKS : CALCULATE TAIL CODES (0,1) AND STORE ON TEMPORARY FILE,   00004910
C                EXCLUDING SAMPLES THAT DO NOT CONTAIN ANY TAIL         00004920
C             STORE TAIL CODES IN Q(), OVERRIDING INITIAL DATA          00004930
C                (IF IDATA = 25, THE TAILS TABLE IS ALREADY STORED)     00004940
C             SAVE (OPTIONAL) VARIABLE LABELS, CUTOFF VALUES, TAILS,    00004950
C                ON UNIT IFT                                            00004960
C             PRINT (OPTIONAL) COMPLETE TAILS TABLE WITH MARGINS        00004970
C             CALL ROUTINES FOR: CALCULATING TABLE MARGINS AND TAIL     00004980
C                                ASSOCIATIONS (DIAGONAL ELEMENTS ONLY), 00004990
C                               PRINTING HISTOGRAM OF SAMPLES' MARGINS. 00005000
C                                                                       00005010
C*****************************************************************     00005020
C MICHEL MELLINGER (SRC), OCT 6/82.                                     00005030
C UPDATES :                                                             00005040
C*****************************************************************     00005050
      COMMON/INOUT/IREAD,IPRT,IDATA,IFT,IFA,IFAST,IFTEMP                00005060
      COMMON/TITLE/TITLE                                                00005070
      COMMON/DIM/NI,NJ,NJ1,NIJ,NJS                                      00005080
      COMMON/PRTSA/LST,LSA,LSAST                                        00005090
      DIMENSION TITLE(20)                                               00005100
      DIMENSION FMT(40)                                                 00005110
      DIMENSION Q(MEMORY),CUTOF(NJ)                                     00005120
      DATA STIP/4H----/                                                 00005130
      DATA PLUS/1H+/                                                    00005140
C                                                                       00005150
C SKIP CALCULATIONS IF TAILS TABLE WAS READ IN INPUT                    00005160
      IF(IDATA.EQ.25) GO TO 200                                         00005170
C                                                                       00005180
C SCAN SAMPLES (LABELS ARE SKIPPED : SEE LL1+1)                         00005190
C                                                                       00005200
C     INITIALIZE SAMPLE COUNTER                                         00005210
         NCOUNT = 0                                                     00005220
C                                                                       00005230
         DO 100 K=1,NI                                                  00005240
            LL1 = K*(NJ + 1)                                            00005250
            LL2 = LL1 + NJ                                              00005260
C                                                                       00005270
C     SCAN DATA VALUES (LL1+1 TO LL1+NJ)                                00005280
C     COMPARE EACH TO CORRESPONDING CUTOFF VALUE                        00005290
C     REPLACE DATA VALUE BY TAIL CODE (0,1)                             00005300
C     CUMULATE TAIL CODES FOR SAMPLE, KEEP SAMPLE ONLY IF IT CONTAINS   00005310
C        AT LEAST ONE TAIL                                              00005320
C                                                                       00005330
C     ...INITIALIZE TAILS COUNTER                                       00005340
            NCJMUL = 0                                                  00005350
```

```
C                                                                       00005360
            DO 10 L=1,NJ                                                00005370
               IF(Q(LL1+L).LT.CUTOF(L)) GO TO 20                        00005380
                  Q(LL1+L) = 1.0                                        00005390
                  NCOMUL = NCOMUL + 1                                   00005400
               GO TO 10                                                 00005410
   20          CONTINUE                                                 00005420
                  Q(LL1+L) = 0.0                                        00005430
   10       CONTINUE                                                    00005440
C                                                                       00005450
C     ...STORE SAMPLE IF SUITABLE (I.E. CONTAINS AT LEAST ONE TAIL)     00005460
            IF(NCOMUL.GE.1) WRITE(IFTEMP)(Q(L),L=LL1,LL2)               00005470
            IF(NCOMUL.GE.1) NCOUNT = NCOUNT + 1                         00005480
C                                                                       00005490
C     SAMPLE CODED AND STORED (IF RELEVANT)                             00005500
  100    CONTINUE                                                       00005510
C                                                                       00005520
C NCOUNT RELEVANT SAMPLES CODED AND STORED                              00005530
C                                                                       00005540
C     PRINT MESSAGE IF SOME SAMPLES HAD NO TAIL                         00005550
         IF(NCOUNT.NE.NI) WRITE(IPRT,1010)NCOUNT                        00005560
C                                                                       00005570
C TRANSFER COMPRESSED TAILS TABLE TO VECTOR Q() LOCATIONS, OVERRIDING    00005580
C  INITIAL DATA                                                         00005590
C                                                                       00005600
C     QUIT IF NO SAMPLES STORED ON IFTEMP (NONE OF THE SAMPLES          00005610
C      HAS A TAIL)                                                      00005620
         IF(NCOUNT.EQ.0) MEMORY=0                                       00005630
         IF(NCOUNT.EQ.0) RETURN                                         00005640
C                                                                       00005650
C     RESET TEMPORARY FILE                                              00005660
         REWIND IFTEMP                                                  00005670
C                                                                       00005680
C     TRANSFER TAIL CODES                                               00005690
         DO 110 K=1,NCOUNT                                              00005700
            LL1 = K*(NJ + 1)                                            00005710
            LL2 = LL1 + NJ                                              00005720
            READ(IFTEMP) (Q(L),L=LL1,LL2)                               00005730
  110    CONTINUE                                                       00005740
C                                                                       00005750
C ALL SAMPLES TRANSFERRED                                               00005760
C RESET Q() PARAMETERS                                                  00005770
         NI = NCOUNT                                                    00005780
         NIJ = NI*(NJ + 1)                                              00005790
C                                                                       00005800
C OPTIONAL SAVING, ON UNIT IFT, OF VARIABLE LABELS,                     00005810
C  CUTOFF VALUES, AND TAILS TABLE (FOR LST=2,3)                         00005820
         IF(LST.LE.1) GO TO 150                                         00005830
C                                                                       00005840
C     SAVE SUCCESSIVELY: VARIABLE LABELS, CUTOFF VALUES, AND TAILS TABLE00005850
            WRITE(IFT,1500) (Q(L),L=1,NJ)                               00005860
            WRITE(IFT,1510) (CUTOF(L),L=1,NJ)                           00005870
            DO 140 K=1,NI                                               00005880
               LL1 = K*(NJ + 1)                                         00005890
               LL2 = LL1 + NJ                                           00005900
               WRITE(IFT,1520) (Q(L),L=LL1,LL2)                         00005910
  140       CONTINUE                                                    00005920
            WRITE(IPRT,1020)                                            00005930
  150    CONTINUE                                                       00005940
```

```
C                                                                       00005950
C PREVIOUS PROCEDURE(FROM FIRST STATEMENT ON) WAS SKIPPED IF TAILS TABLE 00005960
C    WAS ALREADY IN Q(), I.E. FOR IDATA=2>                              00005970
  200 CONTINUE                                                          00005980
C                                                                       00005990
C CHECK MEMORY BEFORE PROCEEDING, QUIT IF CHECK NEGATIVE                00006000
      MEMCHK = NIJ + (2*NJ) + NI                                        00006010
      IF(MEMORY.GE.MEMCHK) GO TO 205                                    00006020
         WRITE(IPRT,1030) MEMORY,MEMCHK                                 00006030
         MEMORY = 0                                                     00006040
         RETURN                                                         00006050
  205 CONTINUE                                                          00006060
C                                                                       00006070
C TAILS TABLE READY ...PROCEED                                          00006080
C                                                                       00006090
C RESET Q() PARAMETERS AND CALCULATE TAILS TABLE MARGINS                00006100
      NF = NJ + NIJ                                                     00006110
      NK = NF + NI                                                      00006120
      NL = NK + NJ                                                      00006130
      CALL MARGIN(Q(NJ+1),Q(NF+1),Q(NK+1),Q(NL+1))                      00006140
C                                                                       00006150
C OPTIONAL PRINTING OF COMPLETE TAILS TABLE                             00006160
      IF(LST.EQ.0.OR.LST.EQ.2) GO TO 300                                00006170
C                                                                       00006180
C     SCAN VARIABLES BY GROUP OF 20                                     00006190
         DO 210 NVAR=1,NJ,20                                            00006200
            NV1 = NVAR                                                  00006210
            NV2 = NV1 + 19                                              00006220
            IF(NV2.GT.NJ) NV2=NJ                                        00006230
            NKNV1 = NK + NV1                                            00006240
            NKNV2 = NK + NV2                                            00006250
            LPAGE = (NVAR+20) / 20                                      00006260
C                                                                       00006270
C     SCAN SAMPLES BY GROUP OF 40                                       00006280
            DO 220 K=1,NI,40                                            00006290
               KK1 = K                                                  00006300
               KK2 = KK1 + 39                                           00006310
               IF(KK2.GT.NI) KK2=NI                                     00006320
C                                                                       00006330
C     ...PRINT TITLE AND TABLE HEADER                                   00006340
               KPAGE = (K+40) / 40                                      00006350
               WRITE(IPRT,1000) TITLE,KPAGE,LPAGE                       00006360
                  WRITE(IPRT,1100)                                      00006370
                  WRITE(IPRT,1110) ((STUP,PLUS),N=NV1,NV2)              00006380
               WRITE(IPRT,1050) (Q(L),L=NV1,NV2)                        00006390
                  WRITE(IPRT,1100)                                      00006400
                  WRITE(IPRT,1110) ((STUP,PLUS),N=NV1,NV2)              00006410
C                                                                       00006420
C     ...PRINT GROUP OF SAMPLES FOR SELECTED VARIABLES                  00006430
               DO 230 KK=KK1,KK2                                        00006440
                  LLAB = KK*(NJ + 1)                                    00006450
                  LV1 = KK*(NJ+1) + NV1                                 00006460
                  LV2 = KK*(NJ+1) + NV2                                 00006470
                  WRITE(IPRT,1050) Q(LLAB),Q(NF+KK),(Q(L),L=LV1,LV2)    00006480
  230          CONTINUE                                                 00006490
  220       CONTINUE                                                    00006500
```

```
C                                                                       00006510
C     PRINT BOTTOM MARGIN FOR SELECTED VARIABLES (ALL SAMPLES)          00006520
C                                                                       00006530
C     ...FIRST CALCULATE TOTAL NUMBER OF TAILS                          00006540
            TOTAL = 0                                                   00006550
            DO 240 K=1,NI                                               00006560
               TOTAL = TOTAL + Q(NF+K)                                  00006570
  240       CONTINUE                                                    00006580
               WRITE(IPRT,1100)                                         00006590
               WRITE(IPRT,1110) ((STIP,PLUS),N=NKNV1,NKNV2)             00006600
            WRITE(IPRT,1070) TOTAL,(Q(L),L=NKNV1,NKNV2)                 00006610
               WRITE(IPRT,1100)                                         00006620
               WRITE(IPRT,1110) ((STIP,PLUS),N=NKNV1,NKNV2)             00006630
  210    CONTINUE                                                       00006640
      GO TO 350                                                         00006650
  300 CONTINUE                                                          00006660
C                                                                       00006670
C PRINT ONLY BOTTOM MARGIN IF TAILS TABLE NOT ASKED FOR                 00006680
C                                                                       00006690
C     PAGE HEADER                                                       00006700
         WRITE(IPRT,1000) TITLE                                         00006710
C                                                                       00006720
C     TABLE HEADER                                                      00006730
         DO 250 NVAR=1,NJ,20                                            00006740
            NV1 = NVAR                                                  00006750
            NV2 = NV1 + 19                                              00006760
            IF(NV2.GT.NJ) NV2=NJ                                        00006770
            NKNV1 = NK + NV1                                            00006780
            NKNV2 = NK + NV2                                            00006790
               WRITE(IPRT,1100)                                         00006800
               WRITE(IPRT,1110) ((STIP,PLUS),N=NV1,NV2)                 00006810
            WRITE(IPRT,1050) (Q(L),L=NV1,NV2)                           00006820
               WRITE(IPRT,1100)                                         00006830
               WRITE(IPRT,1110) ((STIP,PLUS),N=NV1,NV2)                 00006840
C                                                                       00006850
C     CALCULATE TOTAL NUMBER OF TAILS                                   00006860
            TOTAL = 0                                                   00006870
            DO 260 K=1,NI                                               00006880
               TOTAL = TOTAL + Q(NF+K)                                  00006890
  260       CONTINUE                                                    00006900
C                                                                       00006910
C     BOTTOM MARGIN                                                     00006920
               WRITE(IPRT,1100)                                         00006930
               WRITE(IPRT,1110) ((STIP,PLUS),N=NKNV1,NKNV2)             00006940
            WRITE(IPRT,1070) TOTAL,(Q(L),L=NKNV1,NKNV2)                 00006950
               WRITE(IPRT,1100)                                         00006960
               WRITE(IPRT,1110) ((STIP,PLUS),N=NKNV1,NKNV2)             00006970
  250    CONTINUE                                                       00006980
  350 CONTINUE                                                          00006990
C                                                                       00007000
C PRINT HISTOGRAM OF VERTICAL TAILS TABLE MARGIN (ROWS=SAMPLES)         00007010
C ARRAY TK=Q(NF+1, ... ,NK) IS LOST BECAUSE IT IS USED FOR              00007020
C    STORING CELL FREQUENCIES (See SUBROUTINE HISTO)                    00007030
      CALL HISTO(Q(NF+1))                                               00007040
```

```
C                                                                       00007050
C FORMATS                                                               00007060
 1000 FORMAT(1H1///5X,17HTAILS ANALYSIS : ,20A4)                        00007070
 1005 FORMAT(1H1///5X,17HTAILS ANALYSIS : ,20A4//                       00007080
     $10X,5HPAGE ,I3,12H ...SECTION ,I3//)                              00007090
 1010 FORMAT(1H0,5X,39HNO-TAIL SAMPLES HAVE BEEN ELIMINATED : ,I8,      00007100
     $34H SAMPLES REMAIN IN THE TAILS TABLE)                            00007110
 1020 FORMAT(1H0,5X,74HVARIABLE LABELS,CUTOFF VALUES,AND TAILS TABLE HAV00007120
     $E BEEN SAVED ON DISK FILE)                                        00007130
 1030 FORMAT(1H0,6X,52HINSUFFICIENT MEMORY SPACE TO PROCEED : YOU RESERV00007140
     $ED ,I10,13H ...YOU NEED ,I10,//)                                  00007150
 1050 FORMAT(1H ,14H|SAMPLE|CUMUL|,20(A4,1H|))                         00007160
 1060 FORMAT(1X,2H| ,A4,2H |,F5.0,1H|,20(F4.0,1H|))                     00007170
 1070 FORMAT(1H ,5H|TOTAL |,F5.0,1H|,20(F4.0,1H|))                      00007180
 1100 FORMAT(1H ,14H+------+-----+)                                     00007190
 1110 FORMAT(1H+,14X,20(A4,A1))                                         00007200
 1500 FORMAT(20A4/(20A4))                                               00007210
 1510 FORMAT(8E10.4/(8E10.4))                                           00007220
 1520 FORMAT(A4,38F2.0/(40F2.0))                                        00007230
C                                                                       00007240
      RETURN                                                            00007250
      END                                                               00007260
```

```
      SUBROUTINE TAILA(LABEL,T,TL,A,AK,CUTOF)                          00007270
C*****************************************************************************00007280
C SUBROUTINE TAILA CALCULATES TAIL ASSOCIATIONS WITH MODIFIED          00007290
C                  DIAGONAL ELEMENTS                                   00007300
C                                                                      00007310
C    NOTE: DIAGONAL ELEMENTS ARE EQUAL TO THE NUMBER OF ISOLATED TAILS 00007320
C          FOR EACH VARIABLE                                           00007330
C                                                                      00007340
C     LABEL() CONTAINS THE VARIABLE NAMES                              00007350
C     T() IS THE TAILS TABLE                                           00007360
C     TL() IS THE BOTTOM MARGIN (VARIABLES) OF T()                     00007370
C     A() IS THE TAILS ASSOCIATIONS TABLE (UPPER TRIANGULAR FORM)      00007380
C     AK() IS THE MARGIN OF A()                                        00007390
C     CUTOF() CONTAINS THE CUTOFF VALUES USED TO DEFINE THE TAILS      00007400
C                                                                      00007410
C     CALLED BY CALCT (ONLY IF LSA >= 1)                               00007420
C     CALLS LSYMAT                                                     00007430
C           MARGS...                                                   00007440
C           PRTAB...                                                   00007450
C           SATAB                                                      00007460
C                                                                      00007470
C     TASKS : ATTRIBUTE CORRECT LOCATIONS TO DIAGONAL ELEMENTS,        00007480
C                CALCULATED IN SUBROUTINE TAILS(MARGIN)                00007490
C             CALCULATE TAIL ASSOCIATIONS (NON-DIAGONAL ELEMENTS)      00007500
C             CALCULATE MARGIN OF TAIL ASSOCIATIONS TABLE              00007510
C             PRINT AND/OR SAVE RESULTS ON DISK FILE                   00007520
C                                                                      00007530
C*****************************************************************************00007540
C MICHEL MELLINGER (SRC) ,OCT 6/82.                                    00007550
C UPDATES :                                                            00007560
C*****************************************************************************00007570
      COMMON/IOUT/IREAD,IPRT,IDATA,IFT,IFA,IFAST,IFTEMP                00007580
      COMMON/DIM/NI,NJ,NJ1,NIJ,NJS                                     00007590
      COMMON/PRTSA/LST,LSA,LSAST                                       00007600
      DIMENSION LABEL(NJ),T(NJ1,NI),TL(NJ),A(NJS),AK(NJ),CUTOF(NJ)     00007610
C                                                                      00007620
C ATTRIBUTE CORRECT LOCATIONS TO THE NJ DIAGONAL ELEMENTS OF A()       00007630
C (DIAGONAL ELEMENTS WERE CALCULATED BY MARGIN IN TAILS AND STORED     00007640
C  IN LOCATIONS A(1) TO A(NJ) )                                        00007650
C     NOTE: ELEMENT (1,1) IS IN RIGHT LOCATION ALREADY                 00007660
      DO 10 L=2,NJ                                                     00007670
         CALL LSYMAT(L,L,NJ,LL)                                        00007680
         A(LL) = A(L)                                                  00007690
   10 CONTINUE                                                         00007700
```

```
C                                                                       00007710
C CALCULATE TAIL ASSOCIATIONS TABLE (OFF-DIAGONAL ELEMENTS)             00007720
      DO 100 I=2,NJ1                                                    00007730
         II = I - 1                                                     00007740
         DO 110 J=2,NJ1                                                 00007750
            JJ = J - 1                                                  00007760
C                                                                       00007770
C     SKIP LOWER HALF OF (II,JJ) AND DIAGONAL ELEMENTS (II,II)          00007780
            IF(II.GE.JJ) GO TO 110                                      00007790
C                                                                       00007800
C     CALCULATE STRING LOCATION OF (II,JJ) IN A()                       00007810
C     INITIALIZE, THEN CALCULATE MATRIX ELEMENT                         00007820
            CALL LSYMAT(II,JJ,NJ,LOC)                                   00007830
            A(LOC) = 0.0                                                00007840
            DO 120 K=1,NI                                               00007850
               A(LOC) = A(LOC) + (T(I,K)*T(J,K))                        00007860
  120       CONTINUE                                                    00007870
  110    CONTINUE                                                       00007880
  100 CONTINUE                                                          00007890
C                                                                       00007900
C CALCULATE TAIL ASSOCIATIONS MARGINAL FREQUENCY AK()                   00007910
C ( = NUMBER OF TIMES VARIABLE J WAS INVOLVED IN TAIL ASSOCIATIONS)     00007920
      CALL MARGS(A,AK)                                                  00007930
C                                                                       00007940
C PRINT (LSA=1,3) AND/OR SAVE (LSA=2,3) TAIL ASSOCIATIONS,              00007950
C  TABLE(UPPER HALF), AND MARGINS OF INTEREST                           00007960
      IF(LSA.EQ.2) GO TO 250                                            00007970
C                                                                       00007980
C     PRINT                                                             00007990
         CALL PRTAB(LABEL,CUTOF,A,TL,AK,1)                              00008000
  250 CONTINUE                                                          00008010
      IF(LSA.EQ.1.OR.IDATA.EQ.IFA) GO TO 300                            00008020
C                                                                       00008030
C     SAVE                                                              00008040
         CALL SATAB(LABEL,CUTOF,A,TL,IFA)                               00008050
         WRITE(IPRT,1020)                                               00008060
  300 CONTINUE                                                          00008070
      IF(LSA.NE.1.AND.IDATA.EQ.IFA) WRITE(IPRT,1010)                    00008080
C                                                                       00008090
C FORMATS                                                               00008100
 1010 FORMAT(1H0,5X,97HWARNING...TAIL ASSOCIATIONS TABLE WAS NOT SAVED  00008110
     $ECAUSE IDATA=30 (UNIT REQUIRED FOR SAVING TABLE))                 00008120
 1020 FORMAT(1H0,5X,64HTAIL ASSOCIATIONS TABLE AND MARGIN HAVE BEEN SAVE00008130
     $D ON DISK FILE  )                                                 00008140
C                                                                       00008150
      RETURN                                                            00008160
      END                                                               00008170
```

```
      SUBROUTINE NTAIL(LABEL,TL,A,AK,AST,ASTK,CUTOF)                    00008150
C*********************************************************************00008190
C SUBROUTINE NTAIL CALCULATES THE NORMALIZED TAIL ASSOCIATIONS          00008200
C                                                                       00008210
C     NOTE: THE NORMALIZATION IS DONE WITH RESPECT TO THE MINIMUM OF    00008220
C           THE TWO NUMBERS OF SAMPLES INVOLVED FOR THE TWO VARIABLES   00008230
C           CONSIDERED                                                  00008240
C                                                                       00008250
C     LABEL() CONTAINS THE VARIABLE NAMES                               00008260
C     TL() IS THE BOTTOM MARGIN (VARIABLES) OF THE TAILS TABLE          00008270
C     A() IS THE TAIL ASSOCIATIONS TABLE (UPPER TRIANGULAR FORM)        00008280
C     AK() IS THE MARGIN OF A()                                         00008290
C     AST() IS THE NORMALIZED TAIL ASSOCIATIONS TABLE (UPPER            00008300
C           TRIANGULAR FORM)                                            00008310
C     ASTK() CONTAINS THE AVERAGE NUMBER OF TAIL ASSOCIATIONS PER       00008320
C           SAMPLE INVOLVED                                             00008330
C     CUTOF() CONTAINS THE CUTOFF VALUES USED TO DEFINE THE TAILS       00008340
C                                                                       00008350
C     CALLED BY CALCT (ONLY IF LSAST >= 1)                              00008360
C     CALLS LSYMAT                                                      00008370
C           PRTAB...                                                    00008380
C           SATAB                                                       00008390
C                                                                       00008400
C     TASKS : NORMALIZE TAIL ASSOCIATIONS FROM A()                      00008410
C             CALCULATE AVERAGE NUMBER OF TAIL ASSOCIATIONS PER SAMPLE, 00008420
C               FOR EACH VARIABLE                                       00008430
C             PRINT AND/OR SAVE RESULTS                                 00008440
C                                                                       00008450
C*********************************************************************00008460
C MICHEL MELLINGER (SRC), OCT 6/82.                                     00008470
C UPDATES :                                                             00008480
C*********************************************************************00008490
      COMMON/INOUT/IREAD,IPRT,IDATA,IPT,IFA,IFAST,IFTEMP                00008500
      COMMON/QDIM/NI,NJ,NJ1,NIJ,NJS                                     00008510
      COMMON/PRTSA/LST,LSA,LSAST                                        00008520
      DIMENSION LABEL(NJ),TL(NJ),A(NJS),AK(NJ)                          00008530
      DIMENSION AST(NJS),ASTK(NJS),CUTOF(NJ)                            00008540
C                                                                       00008550
C SCAN INDEXES K AND L OF (UPPER TRIANGULAR) TAIL ASSOCIATIONS TABLE     00008560
      DO 10 K=1,NJ                                                      00008570
         DO 20 L=1,NJ                                                   00008580
C                                                                       00008590
C     SKIP ELEMENTS FROM LOWER (OFF-DIAGONAL) PORTION OF TABLE          00008600
            IF(K.GT.L) GO TO 20                                         00008610
C                                                                       00008620
C     LOCATE ELEMENT (K,L) IN STRING A() AND AST(), AND NORMALIZE       00008630
            CALL LSYMAT(K,L,NJ,KL)                                      00008640
            IF(TL(K).LE.0.001.OR.TL(L).LE.0.001) AST(KL)=99999.         00008650
            IF(TL(K).LE.0.001.OR.TL(L).LE.0.001) GO TO 20               00008660
            AST(KL) = A(KL) / ( AMIN1(TL(K),TL(L)) )                    00008670
   20    CONTINUE                                                       00008680
   10 CONTINUE                                                          00008690
```

```
C                                                                       00008700
C CALCULATE AVERAGE NUMBER OF TAIL ASSOCIATIONS PER SAMPLE ,            00008710
C FOR EACH VARIABLE                                                     00008720
      DO 50 K=1,NJ                                                      00008730
         IF(TL(K).LE.0.001) ASTK(K)=99999.                              00008740
         IF(TL(K).LE.0.001) GO TO 50                                    00008750
         ASTK(K) = AK(K) / TL(K)                                        00008760
   50 CONTINUE                                                          00008770
C                                                                       00008780
C MULTIPLY AST() VALUES BY 100 FOR OUTPUT                               00008790
      DO 60 K=1,NJS                                                     00008800
         AST(K) = 100. * AST(K)                                         00008810
   60 CONTINUE                                                          00008820
C                                                                       00008830
C PRINT (LSAST=1,3) AND/OR SAVE (LSAST=2,3) THE RESULTS                 00008840
      IF(LSAST.EQ.2) GO TO 100                                          00008850
C                                                                       00008860
C     PRINT                                                             00008870
         CALL PRTAB(LABEL,CUTOF,AST,TL,ASTK,2)                          00008880
  100 CONTINUE                                                          00008890
      IF(LSAST.EQ.1.OR.IDATA.EQ.IFAST) GO TO 200                        00008900
C                                                                       00008910
C     SAVE                                                              00008920
         CALL SATAB(LABEL,CUTOF,AST,TL,IFAST)                           00008930
         WRITE(IPRT,1020)                                               00008940
  200 CONTINUE                                                          00008950
      IF(LSAST.NE.1.AND.IDATA.EQ.IFAST) WRITE(IPRT,1010)                00008960
C                                                                       00008970
C FORMATS                                                               00008980
 1010 FORMAT(1HO,5X,107HWARNING...NORMALIZED TAIL ASSOCIATIONS TABLE WAS 00008990
     $NOT SAVED BECAUSE IDATA=35 (UNIT REQUIRED FOR SAVING TABLE))      00009000
 1020 FORMAT(1HO,5X,87HNORMALIZED TAIL ASSOCIATIONS TABLE AND MARGIN OF 00009010
     $INTEREST HAVE BEEN SAVED ON DISK FILE  )                          00009020
C                                                                       00009030
      RETURN                                                            00009040
      END                                                               00009050
```

```
      SUBROUTINE CONV(Q,MEMORY)                                          00009060
C*********************************************************************** 00009070
C SUBROUTINE CONV CONVERTS SPECIAL CODES CONTAINED IN THE DATA ,          00009080
C                   ACCORDING TO USER-SPECIFIED RULES                     00009090
C                                                                         00009100
C     CALLED BY DATAIN                                                    00009110
C                                                                         00009120
C     TASKS : THE ROUTINE PROVIDED HERE CONVERTS SPECIAL CODES FROM       00009130
C             THE SRC'S LITHOGEOCHEMISTRY MASTER FILE (SEE SRC PUBLI-     00009140
C              CATION# R-740-1-E-83, M.MELLINGER) :                       00009150
C                                                                         00009160
C               VALUE  MEANING                  NEW VALUE               00009170
C               -----  -------                  ---------               00009180
C               9999   ANALYSIS NOT POSSIBLE       0                    00009190
C                <0    TRACE ELT IN % RANGE     ABS(OLD VALUE*1000)      00009200
C                                                                         00009210
C*********************************************************************** 00009220
C MICHEL MELLINGER (SRC), OCT 6/82.                                       00009230
C UPDATES :                                                               00009240
C*********************************************************************** 00009250
      COMMON/INOUT/IREAD,IPRT,IDATA,IFT,IFA,IFAST,IFTEMP                  00009260
      COMMON/QDIM/NI,NJ,NJ1,NIJ,NJS                                       00009270
      DIMENSION Q(MEMORY)                                                 00009280
C                                                                         00009290
C CONVERT SPECIAL CODES FOR EACH OF THE NI SAMPLES                        00009300
C                                                                         00009310
C     SET COUNTERS FOR NUMBER OF CONVERSIONS, FOR EACH SPECIAL CODE CASE  00009320
         N9999 = 0                                                        00009330
         NEGAT = 0                                                        00009340
         NTOTL = 0                                                        00009350
      DO 100 K=1,NI                                                       00009360
         LL1 = K*(NJ + 1)                                                 00009370
         LL2 = LL1 + NJ                                                   00009380
C                                                                         00009390
C     SKIP SAMPLE LABEL                                                   00009400
         LL1 = LL1+1                                                      00009410
C                                                                         00009420
C     SCAN DATA FOR SAMPLE K                                              00009430
         DO 200 L=LL1,LL2                                                 00009440
C                                                                         00009450
C     ...TEST FOR VALUES=9999. ,CONVERT ANY TO 0                          00009460
            IF(Q(L).LT.9998.5.OR.Q(L).GT.9999.5) GO TO 10                 00009470
               Q(L) = 0.0                                                 00009480
               N9999 = N9999 + 1                                          00009490
   10       CONTINUE                                                      00009500
C                                                                         00009510
C     ...TEST FOR NEGATIVE VALUES, CONVERT ANY TO ACTUAL CONCENTRATION    00009520
            IF(Q(L).GE.-0.0001) GO TO 20                                  00009530
               Q(L) = ABS(Q(L)*1000.0)                                    00009540
               NEGAT = NEGAT + 1                                          00009550
   20       CONTINUE                                                      00009560
  200    CONTINUE                                                         00009570
C                                                                         00009580
C     END OF SCAN FOR SAMPLE K                                            00009590
  100 CONTINUE                                                            00009600
C                                                                         00009610
C CONVERSIONS COMPLETED, PRINT NUMBER OF CASES CONVERTED, IF ANY          00009620
      NTOTL = N9999 + NEGAT                                               00009630
      IF(NTOTL.GT.0) WRITE(IPRT,1000)NTOTL                                00009640
      IF(N9999.GT.0) WRITE(IPRT,1010)N9999                                00009650
      IF(NEGAT.GT.0) WRITE(IPRT,1020)NEGAT                                00009660
C                                                                         00009670
C FORMATS                                                                 00009680
 1000 FORMAT(1H0,5X,7HCONV...,I3,13H DATA CONVERTED : )                   00009690
 1010 FORMAT(1H+,40X,I3,21H "9999" CODES TO ZERO//)                       00009700
 1020 FORMAT(1H+,40X,I3,31H NEGATIVE CODES TO ACTUAL VALUE)               00009710
C                                                                         00009720
      RETURN                                                              00009730
      END                                                                 00009740
```

```
      SUBROUTINE MARGIN(T,TK,TL,A)                                      00009750
C*********************************************************************  00009760
C SUBROUTINE MARGIN CALCULATES MARGINS AND ISOLATED-TAIL SCORES OF      00009770
C                     THE TAILS TABLE                                   00009780
C                                                                       00009790
C     T() IS THE TAILS TABLE                                            00009800
C     TK() IS THE VERTICAL MARGIN (SAMPLES) OF T()                      00009810
C     TL() IS THE BOTTOM MARGIN (VARIABLES) OF T()                      00009820
C     A() IS THE TAIL ASSOCIATIONS TABLE, USED HERE FOR THE TEMPORARY   00009830
C        STORAGE OF THE DIAGONAL ELEMENTS OF THIS TABLE                 00009840
C                                                                       00009850
C     CALLED BY TAILS                                                   00009860
C                                                                       00009870
C     TASKS : CALCULATE TK MARGIN (VERTICAL)                            00009880
C             CALCULATE ISOLATED-TAIL SCORES FOR EACH VARIABLE          00009890
C                (TEMPORARY LOCATION IN A() )                           00009900
C             CALCULATE TL MARGIN (BOTTOM)                              00009910
C                                                                       00009920
C*********************************************************************  00009930
C MICHEL MELLINGER (SRC), OCT 6/32.                                     00009940
C UPDATES :                                                             00009950
C*********************************************************************  00009960
      COMMON/JDIM/NI,NJ,NJ1,NIJ,NJS                                     00009970
      DIMENSION T(NJ1,NI),TK(NI),TL(NJ),A(NJS)                          00009980
C                                                                       00009990
C CALCULATE TK MARGIN AND ISOLATED-TAIL SCORES                          00010000
C                                                                       00010010
C     FIRST INITIALIZE ARRAYS TK() AND A()                              00010020
      DO 10 K=1,NI                                                      00010030
         TK(K) = 0.0                                                    00010040
   10 CONTINUE                                                          00010050
      DO 20 J=1,NJS                                                     00010060
         A(J) = 0.0                                                     00010070
   20 CONTINUE                                                          00010080
C                                                                       00010090
C     EACH TK() ELEMENT IS THE TOTAL NUMBER OF TAILS FOR EACH SAMPLE    00010100
      DO 50 K=1,NI                                                      00010110
C                                                                       00010120
C     ...SKIP SAMPLE LABEL (L=1)                                        00010130
C     ...ITV = ISOLATED-TAIL INDICATOR                                  00010140
         ITV = 0                                                        00010150
         DO 60 L=2,NJ1                                                  00010160
            TK(K) = TK(K) + T(L,K)                                      00010170
            ITV = ITV + (L-1)*T(L,K)                                    00010180
   60    CONTINUE                                                       00010190
C                                                                       00010200
C     ...IF TK(K)=1, INCREMENT ISOLATED-TAIL COUNTER FOR VARIABLE ITV   00010210
         IF(TK(K).EQ.1.0) A(ITV)=A(ITV) + 1.0                           00010220
   50 CONTINUE                                                          00010230
C                                                                       00010240
C CALCULATE TL MARGIN                                                   00010250
C                                                                       00010260
C     FIRST INITIALIZE ARRAY TL()                                       00010270
      DO 100 L=1,NJ                                                     00010280
         TL(L) = 0.0                                                    00010290
  100 CONTINUE                                                          00010300
C                                                                       00010310
C     EACH TL() ELEMENT IS TOTAL NUMBER OF TAILS FOR EACH VARIABLE      00010320
      DO 150 L=1,NJ                                                     00010330
C                                                                       00010340
C     ...SET CORRECT VARIABLE COLUMN# FOR ARRAY T() (SKIP SAMPLE LABEL) 00010350
         LL = L + 1                                                     00010360
         DO 160 K=1,NI                                                  00010370
            TL(L) = TL(L) + T(LL,K)                                     00010380
  160    CONTINUE                                                       00010390
  150 CONTINUE                                                          00010400
C                                                                       00010410
      RETURN                                                            00010420
      END                                                               00010430
```

```
      SUBROUTINE HISTO(TK)                                                  00010440
C*************************************************************************00010450
C SUBROUTINE HISTO CALCULATES AND PRINTS (LINE PER LINE) THE                00010460
C                    HISTOGRAM OF VECTOR TK()                               00010470
C                                                                           00010480
C     TK() IS THE VERTICAL MARGIN (SAMPLES) OF THE TAILS TABLE              00010490
C                                                                           00010500
C     CALLED BY TAILS                                                       00010510
C                                                                           00010520
C     TASKS : SCAN CELL NUMBERS FROM 1 UP, AND                              00010530
C                COUNT ABSOLUTE FREQUENCY OF CELL; STORE ON                 00010540
C                TEMPORARY FILE                                             00010550
C             TRANSFER CELL SEQUENCE BACK TO VECTOR TK()                    00010560
C             PRINT LINE PER LINE HISTOGRAM                                 00010570
C                                                                           00010580
C     NOTES: 1-BECAUSE CELL SEQUENCE IS STORED BACK INTO VECTOR TK(),       00010590
C               SOME OF THE INITIAL DATA FROM TK() ARE LOST                 00010600
C            2-NUMBER OF CELLS (I.E. MAXIMUM NUMBER OF VARIABLES)           00010610
C               IS LIMITED TO 100 IN THE ROUTINE                            00010620
C            3-CELL ABSOLUTE FREQUENCY LIMITED TO 100 ON PRINTOUT           00010630
C               (ACTUAL VALUE PRINTED IN THIS CASE)                         00010640
C*************************************************************************00010650
C MICHEL MELLINGER (SRC), OCT 6/82.                                         00010660
C UPDATES :                                                                 00010670
C*************************************************************************00010680
      COMMON/TITLE/TITLE                                                    00010690
      COMMON/INOUT/IREAD,IPRT,IDATA,IFT,IFA,IFAST,IFTEMP                    00010700
      COMMON/QDIM/NI,NJ,NJ1,NIJ,NJS                                         00010710
      DIMENSION TITLE(20)                                                   00010720
      DIMENSION TK(NI)                                                      00010730
      DATA STAR/1H*/                                                        00010740
C                                                                           00010750
C INITIALIZE TEMPORARY FILE                                                 00010760
      REWIND IFTEMP                                                         00010770
C                                                                           00010780
C SCAN CELL NUMBER FROM 1 TO NJ (NUMBER OF VARIABLES)                       00010790
C THE MAXIMUM NUMBER OF TAILS PER SAMPLE (I.E. OF VARIABLES PER SAMPLE)     00010800
C     IS SET AT 100 IN THIS ROUTINE. THE ACTUAL NUMBER OF CELLS             00010810
C     IS NCELL=MIN(NJ,100), BELOW, AND MIN(NI,NJ)...SEE FURTHER             00010820
      NCELL = MIN0(NJ,100)                                                  00010830
      DO 10 IC=1,NCELL                                                      00010840
C                                                                           00010850
C     SCAN TK() VALUES FROM BOTTOM UP TO FILL CELL # IC (COUNTER NIC)       00010860
            NIC = 0                                                         00010870
            RIC = IC                                                        00010880
            RIC1 = RIC - 0.1                                                00010890
            RIC2 = RIC + 0.1                                                00010900
            DO 20 I=1,NI                                                    00010910
               IF(TK(I).GT.RIC1.AND.TK(I).LT.RIC2) NIC = NIC + 1            00010920
   20       CONTINUE                                                        00010930
C                                                                           00010940
C     SAVE CELL FREQUENCY ON TEMPORARY FILE                                 00010950
            WRITE(IFTEMP) NIC                                               00010960
   10 CONTINUE                                                              00010970
```

```
C                                                                       00010980
C ALL HISTOGRAM FREQUENCIES CALCULATED, STORE BACK INTO VECTOR IK()      00010990
C  TRANSFER A MAXIMUM OF NI CELLS SO AS NOT TO OVERFLOW IK()            00011000
      REWIND IPTEMP                                                     00011010
      IF(NCELL.GT.NI) NCELL=NI                                          00011020
      DO 100 IC=1,NCELL                                                 00011030
         READ(IPTEMP) NIC                                               00011040
         IK(IC) = NIC                                                   00011050
  100 CONTINUE                                                          00011060
C                                                                       00011070
C PRINT HISTOGRAM OF IK() AFGIN                                         00011080
      WRITE(IPRT,1000) TITLE                                            00011090
      WRITE(IPRT,1010)                                                  00011100
      DO 200 IC=1,NCELL                                                 00011110
C                                                                       00011120
C     IF FREQUENCY LOWER THAN 100, PRINT REQUIRED NUMBER OF STARS       00011130
C     IF FREQUENCY = 0, PRINT ONLY CELL NUMBER                          00011140
C     IF FREQUENCY GREATER THAN 100, SATURATE AND PRINT ACTUAL FREQUENCY 00011150
         NSTAR = IK(IC)                                                 00011160
         IF(NSTAR.EQ.0) GO TO 210                                       00011170
         IF(NSTAR.GE.101) GO TO 211                                     00011180
            WRITE(IPRT,1020) IC,(STAR,K=1,NSTAR)                        00011190
         GO TO 200                                                      00011200
  210    CONTINUE                                                       00011210
            WRITE(IPRT,1020) IC                                         00011220
         GO TO 200                                                      00011230
  211    CONTINUE                                                       00011240
            WRITE(IPRT,1020) IC,(STAR,K=1,100),NSTAR                    00011250
  200 CONTINUE                                                          00011260
            WRITE(IPRT,1030)                                            00011270
C                                                                       00011280
C FORMATS                                                               00011290
 1000 FORMAT(1H1///5X,17HTAILS ANALYSIS : ,20A4//                       00011300
     $10X,31HHISTOGRAM OF SAMPLE TAIL-SCORES//)                         00011310
 1010 FORMAT(1X,42H+------+ABSOLUTE FREQUENCY (SAMPLES)------>/         00011320
     1 1X,107H|SCORE||         10        20        30        40        50 00011330
     2        60        70        80        90       100/              00011340
     3 1X,107H+------+|         |         |         |         |         | 00011350
     4         |         |         |         |         ||                00011360
 1020 FORMAT(1X,1H|,I5,1H|,100A1,I4)                                    00011370
 1030 FORMAT(1X,7H+------+)                                             00011380
C                                                                       00011390
      RETURN                                                            00011400
      END                                                               00011410
```

```
      SUBROUTINE LSYMAT(I,J,NJ,LOC)                                     00011420
C*********************************************************************00011430
C SUBROUTINE LSYMAT CALCULATES THE STRING LOCATION (LOC) OF AN          00011440
C                    ELEMENT (I,J) IN A SYMMETRICAL MATRIX (NJ X NJ)    00011450
C                    STORED AS A STRING IN UPPER TRIANGULAR FORM        00011460
C                                                                       00011470
C METHOD USED : "ZERO-INDEX" ALGORITHM FOR THE UPPER HALF ELEMENTS,     00011480
C                  ADAPTED FROM: H.-M. HAMDEN - JUL AND ADVANCED        00011490
C                  FORTRAN PROGRAMMING - ELSEVIER,1973,P.91-93 .        00011500
C                                                                       00011510
C NOTE...FOR THE CALCULATIONS, INDEXES I AND J ARE SWITCHED IF I>J      00011520
C        THE INITIAL VALUES ARE RE-ASSIGNED AFTER THE CALCULATIONS      00011530
C                                                                       00011540
C     CALLED BY MAPGS (TAILA)                                           00011550
C               PRTAB (TAILA,NTAIL)                                     00011560
C               SATAB (NTAIL)                                           00011570
C                                                                       00011580
C     TASKS : CALCULATE ZERO-INDEX IX FOR EACH ROW                      00011590
C             CALCULATE STRING LOCATION OF DIAGONAL ELEMENTS            00011600
C             CALCULATE STRING LOCATION OF OFF-DIAGONAL ELEMENTS        00011610
C                                                                       00011620
C*********************************************************************00011630
C MICHEL MELLINGER (SMC), OCT 6/82.                                     00011640
C UPDATES :                                                             00011650
C*********************************************************************00011660
C                                                                       00011670
C CONSIDER ALL ELEMENTS AS BELONGING TO THE UPPER HALF OF THE MATRIX     00011680
C  ISWIT IS THE INDICATOR FOR I & J SWITCHING                           00011690
      ISWIT = 0                                                         00011700
      IF(I.LE.J) GO TO 10                                               00011710
         JT = I                                                         00011720
         I = J                                                          00011730
         J = JT                                                         00011740
         ISWIT = 1                                                      00011750
   10 CONTINUE                                                          00011760
C                                                                       00011770
C CALCULATE ZERO-INDEX IX OF ROW I                                      00011780
C (= NUMBER OF MATRIX ELEMENTS PRECEDING DIAGONAL ELEMENT (I,I) )       00011790
      IF(I.EQ.1) IX = 0                                                 00011800
      IF(I.EQ.1) GO TO 30                                               00011810
         IX = NJ                                                        00011820
      IF(I.EQ.2) GO TO 30                                               00011830
         II2 = I - 2                                                    00011840
         DO 20 II=1,II2                                                 00011850
            IX = IX + (NJ-II)                                           00011860
   20    CONTINUE                                                       00011870
   30 CONTINUE                                                          00011880
C                                                                       00011890
C CALCULATE STRING LOCATION OF DIAGONAL ELEMENT (I,I)                   00011900
      LII = IX + 1                                                      00011910
      IF(I.EQ.J) LOC = LII                                              00011920
C                                                                       00011930
C CALCULATE STRING LOCATION OF OFF-DIAGONAL ELEMENT (I,J) ... J > I     00011940
      IF(I.NE.J) LOC = LII + (J-I)                                      00011950
C                                                                       00011960
C RE-ASSIGN INITIAL I & J VALUES IF THEY WERE SWITCHED FOR CALCULATIONS 00011970
      IF(ISWIT.EQ.0) GO TO 50                                           00011980
         JT = I                                                         00011990
         I = J                                                          00012000
         J = JT                                                         00012010
   50 CONTINUE                                                          00012020
C                                                                       00012030
      RETURN                                                            00012040
      END                                                               00012050
```

```
      SUBROUTINE MARGS(TAB,TABK)                                        00012060
C*************************************************************************00012070
C SUBROUTINE MARGS CALCULATES THE MARGIN OF A SYMMETRICAL MATRIX        00012080
C                    STORED IN UPPER HALF TRIANGULAR FORM               00012090
C                                                                       00012100
C     TAB() IS THE UPPER TRIANGULAR MATRIX                              00012110
C     TABK() IS THE MARGIN OF TAB()                                     00012120
C                                                                       00012130
C     CALLED BY TABLA                                                   00012140
C     CALLS LSYMAT                                                      00012150
C                                                                       00012160
C*************************************************************************00012170
C MICHEL MELLINGER (SRC), OCT 6/82.                                     00012180
C UPDATES :                                                             00012190
C*************************************************************************00012200
      COMMON/DDIM/NI,NJ,NJ1,NIJ,NJS                                     00012210
      DIMENSION TAB(NJS),TABK(NJ)                                       00012220
C                                                                       00012230
C INITIALIZE TABK()                                                     00012240
      DO 10 K=1,NJ                                                      00012250
         TABK(K) = 0.0                                                  00012260
   10 CONTINUE                                                          00012270
C                                                                       00012280
C SCAN THE ROWS OF THE MATRIX                                           00012290
      DO 100 K=1,NJ                                                     00012300
C                                                                       00012310
C     SUM ALL ELEMENTS FROM EACH ROW OF TAB()                           00012320
C     ALL ELEMENTS ARE PICKED FROM STRING TAB(UPPER TRIANGULAR MATRIX)  00012330
         DO 110 L=1,NJ                                                  00012340
            CALL LSYMAT(K,L,NJ,LOC)                                     00012350
            TABK(K) = TABK(K) + TAB(LOC)                                00012360
  110    CONTINUE                                                       00012370
  100 CONTINUE                                                          00012380
C                                                                       00012390
      RETURN                                                            00012400
      END                                                               00012410
```

```
      SUBROUTINE PRTAB(LABEL,CUTOF,TAB,TAB1,TAB2,MODE)                  00012420
C*********************************************************************00012430
C SUBROUTINE PRTAB PRINTS THE SYMMETRICAL TABLE (UPPER TRIANGULAR FORM) 00012440
C                  TAB() WITH MARGINS OF INTEREST                       00012450
C                                                                       00012460
C     LABEL() CONTAINS THE VARIABLE NAMES                               00012470
C     CUTOF() CONTAINS THE CUTOFF VALUES USED TO DEFINE THE TAILS       00012480
C     TAB() IS THE UPPER TRIANGULAR MATRIX                              00012490
C     TAB1() IS THE FIRST MARGIN TO BE PRINTED                          00012500
C     TAB2() IS THE SECOND MARGIN TO BE PRINTED                         00012510
C     MODE AFFECTS THE HEADERS AND IS SET IN THE CALL STATEMENT         00012520
C                                                                       00012530
C     CALLED BY TAILA (MODE=1)                                          00012540
C               NTAIL (MODE=2)                                          00012550
C     CALLS LSYMAT                                                      00012560
C                                                                       00012570
C     TASKS : PRINT HEADER (VARIES ACCORDING TO MODE)                   00012580
C             PRINT UPPER TRIANGULAR MATRIX WITH RELEVANT MARGINS:      00012590
C               -MAXIMUM NUMBER OF COLUMNS = 15 PER PAGE                00012600
C               -MAXIMUM NUMBER OF ROWS = 40 PER PAGE                   00012610
C                                                                       00012620
C*********************************************************************00012630
C MICHEL NELLINGER (SFC), OCT 6/82.                                     00012640
C UPDATES :                                                             00012650
C*********************************************************************00012660
      COMMON/TITLE/TITLE                                                00012670
      COMMON/INOUT/IREAD,IPRT,IDATA,IFT,IFA,IFAST,IFTEMP                00012680
      COMMON/DIM/NI,NJ,NJ1,NIJ,NJS                                      00012690
      DIMENSION TITLE(20)                                               00012700
      DIMENSION LABEL(NJ),CUTOF(NJ),TAB(NJS),TAB1(NJ),TAB2(NJ)          00012710
C                                                                       00012720
C BLANK IS USED TO PRINT STARS (IN I5) TO FILL BLANK FIELDS             00012730
      BLANK = 999999.                                                   00012740
C                                                                       00012750
C INITIALIZE PAGE NUMBER FOR ROW BLOCKS AND SCAN BLOCKS OF 40 ROWS      00012760
C    IPAGE IS THE PAGE#                                                 00012770
C    IROWT IS THE # OF THE TOP ROW                                      00012780
C    IROWB IS THE # OF THE BOTTOM ROW                                   00012790
      IPAGE = 0                                                         00012800
      DO 10 IROW=1,NJ,40                                                00012810
         IPAGE = IPAGE + 1                                              00012820
         IROWT = IROW                                                   00012830
         IROWB = IROWT + 39                                             00012840
         IF(IROWB.GT.NJ) IROWB = NJ                                     00012850
C                                                                       00012860
C     INITIALIZE SECTION NUMBER FOR COLUMN BLOCKS,                      00012870
C     AND SCAN BLOCKS OF 15 COLUMNS                                     00012880
C        JPAGE IS THE SECTION#                                          00012890
C        ICOLL IS THE # OF THE LEFTMOST COLUMN                          00012900
C        ICOLR IS THE # OF THE RIGHTMOST COLUMN                         00012910
         JPAGE = 0                                                      00012920
         DO 20 ICOL=1,NJ,15                                             00012930
            JPAGE = JPAGE + 1                                           00012940
            ICOLL = ICOL                                                00012950
            ICOLR = ICOLL + 14                                          00012960
            IF(ICOLR.GT.NJ) ICOLR = NJ                                  00012970
C                                                                       00012980
C     ...IF THE WINDOW (IROWT,IROWB/ICOLL,ICOLR) IS LOCATED IN THE      00012990
C        LOWER HALF OF THE MATRIX, DO NOT PRINT                         00013000
         IF(IROWT.GT.ICOLR) GO TO 20                                    00013010
C                                                                       00013020
```

```
C     ...PRINT TITLE AND HEADER                                                 00013030
      WRITE(IPRT,990) TITLE                                                     00013040
      WRITE(IPRT,1000)                                                          00013050
      IF(MODE.EQ.1) WRITE(IPRT,1010)IPAGE,JPAGE                                 00013060
      IF(MODE.EQ.2) WRITE(IPRT,1020)IPAGE,JPAGE                                 00013070
      IF(MODE.EQ.1) WRITE(IPRT,1015)                                            00013080
      IF(MODE.EQ.2) WRITE(IPRT,1025)                                            00013090
      IF(MODE.EQ.1) WRITE(IPRT,1030)                                            00013100
      IF(MODE.EQ.2) WRITE(IPRT,1130)                                            00013110
      WRITE(IPRT,1040) (LABEL(L),L=ICOLL,ICOLR)                                 00013120
C                                                                               00013130
C     ...PRINT MATRIX ELEMENTS                                                  00013140
C        NBLKP IS THE NUMBER OF BLANKS TO BE PRINTED AT THE LEFT                00013150
C        NWIDE IS THE TOTAL NUMBER OF FIELDS IN THE ROW                         00013160
C        THE DIAGONAL ELEMENTS ARE RETRIEVED FROM TAB() USING                   00013170
C              SUBROUTINE LSYMAT                                                00013180
C        KK, NLEFT & NRITE DEFINE THE SEQUENCE OF ELEMENTS FROM                 00013190
C              TAB() TO BE PRINTED, FOR THE BLOCK CONSIDERED                    00013200
C                                                                               00013210
C                                                                               00013220
C     ......SCAN ROWS FOR THE BLOCK TO BE PRINTED                               00013230
      DO 30 K=IFOWT,IFOWB                                                       00013240
         NBLKP = K - ICOLL                                                      00013250
         NWIDE = (ICOLR-ICOLL) + 1                                              00013260
         CALL LSYMAT(K,K,NJ,KK)                                                 00013270
         NLEFT = KK + (ICOLL-K)                                                 00013280
         NRITE = NLEFT + (NWIDE-1)                                              00013290
C                                                                               00013300
C     ......IDENTIFY SHAPE OF BLOCK TO BE PRINTED ROW BY ROW,                   00013310
C           AND PRINT TABLE ACCORDINGLY                                         00013320
         IF(K.LE.ICOLL) WRITE(IPRT,1500)                                        00013330
     $    LABEL(K),CUTOF(K),(TAB(L),L=NLEFT,NRITE)                             00013340
         IF(K.GT.ICOLL.AND.K.LT.ICOLR) WRITE(IPRT,1500)                         00013350
     $    LABEL(K),CUTOF(K),(BLANK,L=1,NBLKP),(TAB(L),L=KK,NRITE)              00013360
         IF(K.EQ.ICOLR) WRITE(IPRT,1500)                                        00013370
     $    LABEL(K),CUTOF(K),(BLANK,L=1,NBLKP),TAB(KK)                          00013380
         IF(K.GT.ICOLR) WRITE(IPRT,1500)                                        00013390
     $    LABEL(K),CUTOF(K),(BLANK,L=1,NWIDE)                                  00013400
C                                                                               00013410
C     ......OVERPRINT TAB1 AND TAB2 MARGINS FOR CURRENT ROW                     00013420
            WRITE(IPRT,1510) TAB1(K),TAB2(K)                                    00013430
   30    CONTINUE                                                               00013440
   20 CONTINUE                                                                  00013450
   10 CONTINUE                                                                  00013460
C                                                                               00013470
C FORMATS                                                                       00013480
  990 FORMAT(1H1///5X,17HTAILS ANALYSIS : ,20A4)                                00013490
 1000 FORMAT(1HO,5A,26H* TAIL ASSOCIATIONS TABLE )                              00013500
 1010 FORMAT(1H+,31X,7H(PAGE ,I2,9H,SECTION ,I2,1H))                            00013510
 1015 FORMAT(1HO,5X,51HNOTE: DIAGONAL ELEMENTS ARE EQUAL TO THE NUMBER 00013520
     $F /6X,50H       SAMPLES IN WHICH A VARIABLE HAD AN ISOLATED TAIL.)00013530
 1020 FORMAT(1H+,31X,13H(NORMALIZED)*(PAGE ,I2,9H,SECTION ,I2,1H))      00013540
 1025 FORMAT(1HO,5X,75HNOTE: NORMALIZED TAIL ASSOCIATIONS ARE SIMILARITY00013550
     1 INDEXES BETWEEN VARIABLES/12X,82HDIAGONAL ELEMENTS ARE 'SELF-SIMI00013560
     2LARITY' INDEXES (EXCLUSIVE OF ALL OTHER VARIABLES)/,2X,32HALL SIMI00013570
     3LARITY INDEXES ARE X 100  )                                       00013580
 1030 FORMAT(1H ,100X,23H   NUMBER OF      NUMBER OF  /                 00013590
     1       1H ,100X,23H    SAMPLES          TAIL    /                00013600
     2       1H ,100X,23H   INVOLVED     ASSOCIATIONS)                00013610
 1130 FORMAT(1H ,100X,32H  NUMBER OF      AVERAGE NUMBER /             00013620
     1       1H ,100X,32H   SAMPLES      OF TAIL ASSOCTNS/            00013630
     2       1H ,100X,32H  INVOLVED        PER SAMPLE   )             00013640
 1040 FORMAT(1H+,12X,13HCUTOFF VALUE ,15(1X,A4))                        00013650
 1500 FORMAT(1X,A4,2X,1H(,F10.4,3H)  ,15F5.0)                           00013660
 1510 FORMAT(1H+,130X,5A,F5.0,9X,F6.1)                                  00013670
C                                                                       00013680
      RETURN                                                            00013690
      END
```

```
      SUBROUTINE SATAB(LABEL,CUTOF,TAB,TABI,LUN)                        00013700
C*************************************************************************00013710
C SUBROUTINE SATAB SAVES TAIL ASSOCIATIONS TABLE (NORMALIZED OR NOT,      00013720
C                    UPPER TRIANGULAR FORM) AND INFORMATION OF INTEREST   00013730
C                    ON UNIT # LUN                                        00013740
C                                                                         00013750
C     LABEL() CONTAINS THE VARIABLE NAMES                                 00013760
C     CUTOF() CONTAINS THE CUTOFF VALUES USED TO DEFINE THE TAILS         00013770
C     TAB() IS THE UPPER TRIANGULAR MATRIX                                00013780
C     TABI() IS THE TABLE MARGIN SELECTED IN THE CALL STATEMENT           00013790
C     LUN IS THE LOGICAL UNIT NUMBER OF THE UNIT ON WHICH THE             00013800
C        DATA ARE SAVED                                                   00013810
C                                                                         00013820
C     CALLED BY TAILA                                                     00013830
C               NTAIL                                                     00013840
C                                                                         00013850
C*************************************************************************00013860
C MICHEL MELLINGER (SRC), OCT 6/82.                                       00013870
C UPDATES                                                                 00013880
C*************************************************************************00013890
      COMMON/DDIM/NI,NJ,NJ1,NIJ,NJS                                       00013900
      DIMENSION LABEL(NJ),CUTOF(NJ),TAB(NJS),TABI(NJ)                     00013910
C                                                                         00013920
C SAVE IN ORDER: VARIABLE LABELS, CUTOFS, TAIL ASSOCIATION TABLE(UPPER    00013930
C  TRIANGULAR FORM), AND MARGIN RELATED TO THE (AVERAGE) NUMBER OF        00013940
C  SAMPLES INVOLVED PER VARIABLE                                          00013950
      WRITE(LUN,1500) (LABEL(L),L=1,NJ)                                   00013960
      WRITE(LUN,1510) (CUTOF(L),L=1,NJ)                                   00013970
      WRITE(LUN,1520) (TAB(L),L=1,NJS)                                    00013980
      WRITE(LUN,1520) (TABI(L),L=1,NJ)                                    00013990
C                                                                         00014000
C FORMATS                                                                 00014010
 1500 FORMAT(20A4/(20A4))                                                 00014020
 1510 FORMAT(8E10.4/(8E10.4))                                             00014030
 1520 FORMAT(16F5.0/(16F5.0))                                             00014040
C                                                                         00014050
      RETURN                                                              00014060
      END                                                                 00014070
```

```
*****************************************************************************
*                                                                           *
*   T A I L A N  (TAILS ANALYSIS)                          * INSTRUCTIONS *
*                                                                           *
*****************************************************************************
*
** FUNCTION **
*
*     PROGRAM TAILAN IS A TAILS ANALYSIS PROGRAM THAT
*        INCLUDES AS OPTIONS:
*        - THE COMPLETE SEQUENCE OF CALCULATIONS FROM THE RAW DATA
*        - THE CALCULATION OF THE (NORMALIZED) TAIL ASSOCIATIONS
*           FROM THE TAILS TABLE
*        - THE SAVING OF RESULTS ON FILES FOR LATER USE
*
** DIMENSION **
*
*     MEMORY = NI*(NJ + 1) + NJ**2 + 6*NJ + NI  ,
*       WITH : NI = NUMBER OF INDIVIDUALS(ROWS)
*              NJ = NUMBER OF VARIABLES(COLUMNS)
*
** TEMPORARY WORKING FILE **
*
*     UNIT 15 : BINARY
*
** OUTPUT FILES (IF REQUESTED IN GENERAL PARAMETERS) **
*
*     UNIT 25 : TAILS TABLE
*     UNIT 30 : TAIL ASSOCIATIONS TABLE
*     UNIT 35 : NORMALIZED TAIL ASSOCIATIONS TABLE
*
** PARAMETER CARDS **
*
*     1-TITLE - FORMAT 20A4
*     2-GENERAL PARAMETERS - FORMAT 6I4
*        NI     : NUMBER OF INDIVIDUALS (ROWS) IN THE INPUT DATA
*                 (NOTE : ...IF NI=0 , THE PROGRAM WILL COUNT THE NUMBER
*                  OF INDIVIDUALS IN THE INPUT FILE, TO THE EOF)
*        NJ     : NUMBER OF VARIABLES  (COLUMNS) IN THE INPUT DATA
*                 (NOTE : NJ MUST BE GREATER THAN 1)
*        IDATA  : LOGICAL UNIT NUMBER ON WHICH THE DATA ARE READ
*                 (NOTE : ...IF IDATA=25, THE TAILS TABLE IS READ AS INPUT
*                  ON UNIT # 25)
*        IFMT   : NUMBER OF CARDS USED TO DEFINE THE DATA INPUT FORMAT
*                   =0 ASSUMES THE TAILS TABLE IS READ AS INPUT ON UNIT#25
*                   =1 DATA INPUT FORMAT GIVEN ON ONE 20A4 CARD
*                   =2 DATA INPUT FORMAT GIVEN ON TWO 20A4 CARDS
*        ICONV  : =0 THE INPUT DATA DO NOT REQUIRE RE-CODING OF SPECIAL
*                     CODES PRIOR TO CALCULATIONS
*                 =1 THE INPUT DATA REQUIRE RE-CODING OF SPECIAL CODES
*                     PRIOR TO CALCULATIONS
*                 (NOTE : THE PROGRAM INCLUDES A CONVERSION ROUTINE FOR
*                  NEGATIVE CODES ACCORDING TO THE 'SAC LITHOGEOCHEMICAL
*                  DATA FILE' CODING RULES. IF YOU REQUIRE ANOTHER
*                  RE-CODING SCHEME, YOU MUST PROVIDE YOUR OWN FORTRAN
*                  ROUTINE ...SEE USER MANUAL)
```

```
*        LST   : =0 PRINT THE BOTTOM MARGIN (VARIABLES/COLUMNS) FROM    *00000570
*                   THE TAILS TABLE, PLUS THE HISTOGRAM OF THE          *00000580
*                   VERTICAL MARGIN OF THE TAILS TABLE(INDIVIDUALS/     *00000590
*                   ROWS)                                               *00000600
*                 =1 PRINT THE TAILS TABLE WITH BOTTOM (VARIABLES/      *00000610
*                   COLUMNS) AND VERTICAL (INDIVIDUALS/ROWS) MARGINS,   *00000620
*                   PLUS THE HISTOGRAM OF THE VERTICAL MARGIN.          *00000630
*                 =2 SAVE THE VARIABLE LABELS, CUTOFF VALUES, AND TAILS *00000640
*                   TABLE ON LOGICAL UNIT # 25                          *00000650
*                 =3 CUMULATE OPTIONS 1 AND 2 ABOVE                     *00000660
*        LSA   : =0 DO NOT CALCULATE TAIL ASSOCIATIONS                  *00000670
*                 =1 PRINT TAIL ASSOCIATIONS TABLE WITH MARGINS OF      *00000680
*                   INTEREST                                            *00000690
*                 =2 SAVE TAIL ASSOCIATIONS TABLE WITH MARGIN OF        *00000700
*                   INTEREST, ON LOGICAL UNIT # 30                      *00000710
*                   LOGICAL UNIT # 30                                   *00000720
*                 =3 CUMULATE OPTIONS 1 AND 2 ABOVE                     *00000730
*        LSAST : =0 DO NOT CALCULATE NORMALIZED TAIL ASSOCIATIONS       *00000740
*                 =1 PRINT NORMALIZED TAIL ASSOCIATIONS TABLE WITH      *00000750
*                   MARGINS OF INTEREST                                 *00000760
*                 =2 SAVE NORMALIZED TAIL ASSOCIATIONS TABLE WITH       *00000770
*                   MARGIN OF INTEREST, ON LOGICAL UNIT # 35            *00000780
*                 =3 CUMULATE OPTIONS 1 AND 2 ABOVE                     *00000790
*     3-DATA INPUT FORMAT - FORMAT IFMT*(20A4)                          *00000800
*        ...NOT REQUIRED IF IDATA = 25                                  *00000810
*        FOR EACH INDIVIDUAL, THE FORMAT SHALL READ:                    *00000820
*         - THE IDENTIFIER ...FORMAT A (OUTPUT AS A4)                   *00000830
*         - THE NJ VARIABLES ...FORMAT F                                *00000840
*     4-NAMES OF THE NJ VARIABLES (COLUMNS) - FORMAT 20A4               *00000850
*        ...NOT REQUIRED IF IDATA = 25                                  *00000860
*        (AS MANY CARDS AS REQUIRED BY THE VALUE OF NJ ARE READ)        *00000870
*     5-CUTOFF VALUES FOR THE NJ VARIABLES - FREE FORMAT                *00000880
*        ...NOT REQUIRED IF IDATA = 25                                  *00000890
*        ENTER CUTOFF VALUES IN THE SAME ORDER AS THE VARIABLES, USING  *00000900
*         A COMMA AS DELIMITER BETWEEN SUCCESSIVE VALUES                *00000910
*                                                                       *00000920
** INPUT DATA TABLE **                                                 *00000930
*                                                                       *00000940
*    (6)-DATA TABLE - FORMAT AS GIVEN IN CARD(S) #3 ABOVE               *00000950
*        THE DATA ARE READ ON LOGICAL UNIT NUMBER 'IDATA'               *00000960
*        THE DATA TABLE IS READ ONE ROW AT A TIME                       *00000970
*                                                                       *00000980
** OUTPUT FILES **                                                     *00000990
*                                                                       *00001000
*     UNIT 25 FOR THE TAILS TABLE - CONTENT (FORMAT):                   *00001010
*             NJ VARIABLE LABELS (20A4/(20A4))                          *00001020
*             NJ CUTOFF VALUES (8E10.4/(8E10.4))                        *00001030
*             TAILS TABLE, NI ROWS (A4,38F2.0/(40F2.0))                 *00001040
*     UNIT 30 FOR THE TAIL ASSOCIATIONS TABLE - CONTENT (FORMAT):       *00001050
*             NJ VARIABLE LABELS (20A4/(20A4))                          *00001060
*             NJ CUTOFF VALUES (8E10.4/(8E10.4))                        *00001070
*             TAIL ASSOCIATIONS, UPPER TRIANGULAR TABLE ONLY            *00001080
*               (16F5.0/(16F5.0))                                       *00001090
*             TABLE MARGIN (16F5.0/(16F5.0))                            *00001100
*     UNIT 35 FOR THE NORMALIZED TAIL ASSOCIATIONS TABLE :              *00001110
*        CONTENT AND FORMAT AS FOR TAIL ASSOCIATIONS TABLE (ABOVE)      *00001120
*                                                                       *00001130
```

```
*********************************************************************** *00001110
*                                                                       *00001120
*                                                                       *00001130
** JCL CARDS **                                                         *00001140
*                                                                       *00001150
*     //    JOB  'YOUR PARAMETERS'                                      *00001160
*     //    EXEC PGM=TAILAN,PASS='YOUR PASSWORD'                        *00001170
*     //STEPLIB DD UNIT=USER,DISP=SHR,DSN=SKG100.TAILAN.LOAD(TAILAN)    *00001180
*     //FT05F001 DD *                                                 *00001190
*     = = = = PARAMETER CARDS HERE = = = =                            *00001200
*     = = FOLLOWED BY DATA IF IDATA=5  = =                            *00001210
*     //FT06F001 DD SYSOUT=*                                          *00001220
*     //FT15F001 DD UNIT=USER,SPACE=(6233,(200,50)),                  *00001230
*     //       DCB=(RECFM=VBS,LRECL=6229,BLKSIZE=6233)                *00001240
*     = = = = DEFINE OUTPUT UNITS AS REQUIRED = = = =                 *00001250
*     //                                                              *00001260
*                                                                     *00001270
*   NOTES: IN CARD //FT05F001..., YOU MAY POINT TO A DISK FILE        *00001280
*            CONTAINING THE PARAMETER CARDS                           *00001290
*          IF THE DATA DO NOT FOLLOW THE PARAMETER CARDS IN THE       *00001300
*            INPUT FILE, ADD A CARD //FTXXF001..., POINTING TO A      *00001310
*            DISK FILE WHICH CONTAINS THE DATA TABLE, AND SET         *00001320
*            IDATA=XX IN PARAMETER CARD #2                            *00001330
*                                                                     *00001340
*********************************************************************** *00001350
*                                                                       *00001360
** INSTRUCTIONS REVISED M.M. (DEC. 1/83) **                            *00001370
*                                                                       *00001380
*********************************************************************** *00001390
```

POOR

QUALITY

ORIGINAL